Multifidelity Approaches for Design Under Uncertainty

by

Leo Wai-Tsun Ng

Bachelor of Applied Science, University of Toronto (2007)
Master of Science, Massachusetts Institute of Technology (2009)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aerospace Computational Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2013

Author . . . .
Department of Aeronautics and Astronautics
May 9, 2013

Certified by . . . .
Karen E. Willcox
Professor of Aeronautics and Astronautics
Chair, Thesis Committee

Certified by . . . .
Youssef M. Marzouk
Associate Professor of Aeronautics and Astronautics
Member, Thesis Committee

Certified by . . . .
Mark Drela
Professor of Aeronautics and Astronautics
Member, Thesis Committee

Accepted by . . . .
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Multifidelity Approaches for Design Under Uncertainty

by

Leo Wai-Tsun Ng

Submitted to the Department of Aeronautics and Astronautics
on May 9, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aerospace Computational Engineering

Abstract

Uncertainties are present in many engineering applications and it is important to account for their effects during engineering design to achieve robust and reliable systems. One approach is to represent uncertainties as random inputs to the numerical model of the system and investigate the probabilistic behaviour of the model outputs. However, performing optimization in this setting can be computationally expensive, requiring many evaluations of the numerical model to compute the statistics of the system metrics, such as the mean and the variance of the system performance. Fortunately, in many engineering applications, there are one or more lower fidelity models that approximate the original (high-fidelity) numerical model at lower computational costs. This thesis presents rigorous multifidelity approaches to leverage cheap low-fidelity models and other approximations of the expensive high-fidelity model to reduce the computational expense of optimization under uncertainty.

Solving an optimization under uncertainty problem can require estimates of the statistics at many different design points, incurring a significant number of expensive high-fidelity model evaluations. The multifidelity estimator is developed based on the control variate method to reduce the computational cost of achieving a specified root mean square error in the statistic estimate by making use of the correlation between the outputs of the expensive high-fidelity model and the outputs of the cheap low-fidelity model. The method optimally relegates some of the computational load to the low-fidelity model based on the relative model evaluation cost and the strength of the correlation. It has demonstrated 85% computational savings in an acoustic horn robust optimization example.

When the model is sufficiently smooth in the design space in the sense that a small change in the design variables produces a small change in the model outputs, it has an autocorrelation structure that can be exploited by the control variate method. The information reuse estimator is developed to reduce the computational cost of achieving a specified root mean square error in the statistic estimate by making use of the correlation between the high-fidelity model outputs at one design point and those at a previously visited design point. As the optimization progresses towards the optimum in the design space, the steps taken in the design space often become shorter, increasing the correlation and making the information reuse estimator more efficient. To further reduce the computational cost, the combined estimator is developed to incorporate the features of both the multifidelity estimator and the information reuse estimator. It has demonstrated 90% computational savings in the acoustic horn robust optimization example.

The methods developed in this thesis are applied to two practical aerospace applications. In conceptual aircraft design, there are often uncertainties about the future developments of the underlying technologies. The information reuse estimator can be used to efficiently generate a Pareto front to study the trade off between the expected performance and the risk induced by the uncertainties in the different aircraft designs. In a large-scale wing robust optimization problem with uncertainties in material properties and flight conditions, the combined estimator demonstrated a reasonable solution turnaround time of 9.7 days on a 16-processor desktop machine, paving the way to a larger scale wing optimization problem with distributed uncertainties to account for degradation or damage.

Chair, Thesis Committee: Karen E. Willcox
Title: Professor of Aeronautics and Astronautics

Member, Thesis Committee: Youssef M. Marzouk
Title: Associate Professor of Aeronautics and Astronautics

Member, Thesis Committee: Mark Drela
Title: Professor of Aeronautics and Astronautics

Acknowledgments

I would like to thank my advisor Professor Karen Willcox for her guidance and support over the last four years. She has been an indispensable source of advice both within and outside of academic research. Furthermore, she fosters a friendly and collaborative environment among her students that has help made my experience enjoyable and rewarding. I look forward to capping it off with a tremendous finish for Team Gradient Ascent at the Boston Tough Mudder 2013.

I would like to extend my gratitude to my committee members Professor Youssef Marzouk and Professor Mark Drela. Their advice and suggestions have been invaluable to my research. I also appreciate the useful and insightful comments from my thesis readers Professor Qiqi Wang and Dr. Michael Eldred.

Throughout my six years at MIT, I met many great friends that have made this some of the best years of my life. I am grateful to my former GTL friends Andreas, Björn, François, and Jon for hanging around and their camaraderie. I have also enjoyed the company and much need distractions of my officemates Andrew, Kenji the Husky, and Chelsea.

Finally, I would like to acknowledge support from the Air Force Office of Scientific Research MURI program on Uncertainty Quantification under grant FA9550-09-0613, program manager F. Fahroo. I would also like to thank the Singapore University of Technology and Design for the opportunity to visit and enrich my experience. Lastly, I appreciate the support from the Sandia National Laboratories Computer Science Research Institute and Dr. Michael Eldred for the rewarding internship experience.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Accounting for uncertainty and variability in engineering design is important to achieve robust and reliable systems, but is often too computationally expensive to formalize in an optimization setting. A key challenge is to repeatedly propagate uncertainties from the inputs of a numerical model to its outputs during the course of optimization. We propose multifidelity approaches to lower the computational cost by offloading some of the computational burden to inexpensive surrogates in a rigorous manner. We demonstrate the methods in an aircraft conceptual design problem and a large-scale wing robust optimization problem.

In this chapter, we begin by motivating the need for optimization under uncertainty in §1.1. We establish the notations and the general problem setup for the rest of the thesis in §1.2 and §1.3, respectively. In §1.4, we review the literature on multifidelity methods for optimization and uncertainty quantification as well as developments based on the control variate estimator variance reduction technique, which forms the basis of our approach. Based on the review and the open challenges, we present the thesis objectives in §1.5. We end the chapter with the thesis outline in §1.6

1.1 Motivation and Context

Uncertainties are present in many engineering applications and it is often desirable to quantify and manage the effects of the uncertainties on the system behaviour during the design process. Additional sources of uncertainties are introduced when the physical system is modelled mathematically and when the mathematical model is solved numerically. There are many possible sources of uncertainties, and one classification of uncertainties consists of [26]:

parameter uncertainty: unavailability of the exact values of some of the inputs to the mathematical model;

model inadequacy: accuracy of the mathematical model with respect to the true physics;

numerical uncertainty: numerical errors and approximations in the implementation of the computer model;

parametric variability: inherent variability of some of the inputs to the mathematical model;

observation error: unavoidable variability of experimental measurements; and

interpolation uncertainty: lack of outputs for all possible inputs due to limited resources.

Alternatively, uncertainties may also be be classified as [38]:

aleatoric: irreducible uncertainty due to the intrinsic randomness of the physical system; and

epistemic: reducible uncertainty due to the lack of knowledge about the system or the model.

In the application problems described in this thesis, we focus on parameter uncertainties (e.g., the yield stress of an advanced composite material that has yet to be thoroughly tested) and parametric variability (e.g., flight conditions experienced by an aircraft). A system designed without consideration of the uncertainties can have unexpected results when the actual system is realized—performance may be worse than predicted and requirements may not be satisfied. To properly account for uncertainties during the design process, we turn to methods for optimization under uncertainty.

A computationally inexpensive and simple approach that is commonly applied is to take conservative estimates of parameters and include safety factors in the analysis. The disadvantage is that the results may be overly conservative, the values of the safety factors are ambiguous, and, more dangerously, there is the potential to take the values of these parameters and factors at face value without recognizing that they are simply a crude attempt to account for uncertainties. More rigorous treatments of uncertainties include probability theory, possibility theory [12, 53], and Dempster-Shafer evidence theory [9, 45]. In this work, we represent uncertainties as random variables with probability distributions based on the quantification of the uncertainties. Therefore, we are concerned with the efficient computation of the statistics of system metrics of interest during optimization.

An optimization problem in the presence of uncertainties can be formulated in a variety of ways depending on the context. For example, uncertainties may affect the reliability of the system—the degree to which the system performs its intended functions over a specified interval of time [22]. Therefore, one approach in reliability-based design is to formulate the optimization problem with a constraint on the probability of system failure. Uncertainties may also affect the robustness of the system—the insensitivity of the system to variations [39]. The formulation of the robust optimization problem depends on the choice of robustness measure—for uncertainties represented by random variables, the choices include the expectancy measure, the probabilistic threshold measure, and the statistical feasibility measure [3]. For example, based on the expectancy measure, the objective function to be minimized may be the mean of a loss metric plus the standard deviation of the loss metric. In this formulation, the variance of the loss metric represents the lack of robustness and a balance between the expected loss and the lack of robustness is minimized. A drawback to optimization under uncertainty based on a probabilistic representation of the uncertainties is the significant computational cost needed to propagate the probability distributions of the uncertain input parameters to the outputs of the numerical model of the system. To address the computational expense of optimization under uncertainty, we make use of surrogate models that provide approximate information at a lower computational cost than the original model.

In many engineering applications, a range of numerical models—from cheap low-fidelity models to expensive high-fidelity models—are available to predict the same physical output of interest. Examples of low-fidelity models include low-order simplifications of the underlying physics, less accurate numerical implementations of the governing equations, reduced-order compression of the numerical model, data-fit emulators, etc. We consider a hierarchy of models—that is, we trust the output of one model more than the other (hence high-fidelity versus low-fidelity). Therefore, we develop multifidelity approaches that seek the high-fidelity model solution of the optimization under uncertainty problem; the low-fidelity model accelerates the process but is not intended to alter the solution nor does it replace the high-fidelity model. A common theme in multifidelity methods is that a significant portion of the computation is relegated to the inexpensive low-fidelity model with only occasional recourse to the expensive high-fidelity model as a check or a correction. The key to the effectiveness of most multifidelity methods is that the low-fidelity model outputs have similar trends to the high-fidelity model outputs; in many cases, the absolute errors of the low-fidelity model outputs with respect to the high-fidelity model outputs are irrelevant. For example, in optimization, it is usually beneficial for the gradients of the high-fidelity model to be well-approximated by the gradients of the low-fidelity model [1]. In uncertainty propagation, for the methods we develop in this thesis, we require that the random outputs of the low-fidelity model are correlated with the random outputs of the high-fidelity model when induced by the same random inputs representing the uncertainties.

1.2 General Notations

We begin by defining notations that are relevant to the developments in Chapter 2 and Chapters 3. Let $(\Omega, \mathcal{F}, P)$ be the probability space, where Ω is the sample space, $\mathcal{F}$ is the set of events, and P is the probability measure. We define the random variables $A(\omega)$, $B(\omega)$, $C(\omega)$, $D(\omega)$ and $U(\omega)$ for $\omega \in \Omega$ and their realizations are denoted as a_i, b_i, c_i, d_i, and u_i for the i^{th} sample. We are interested in the statistics of the random variables, such as their means and their variances. We define the exact statistic of interest as s subscripted by the random variable, e.g. s_A, s_B, etc., and the estimator of the statistic as $\hat{s}$ or $\tilde{s}$ subscripted by the random variable, e.g. $\hat{s}_A$, $\tilde{s}_C$, etc. If necessary, we also include as a subscript the computational effort to compute the estimator. As an example, $\hat{s}_{A,p}$ is an estimator of the exact statistic s_A that costs p units of computational effort. For the special case of regular Monte Carlo estimators, we use the typical sample mean notation, e.g. $\bar{a}$, $\bar{d}$, etc.

We define models as $M(\mathbf{x}, \mathbf{u})$ with two inputs: design variables $\mathbf{x}$ and input parameters $\mathbf{u}$. When we need to differentiate between high-fidelity models and low-fidelity models, we use $M_{\text{high}}(\mathbf{x}, \mathbf{u})$ and $M_{\text{low}}(\mathbf{x}, \mathbf{u})$, respectively. We reserve n for the number of high-fidelity model evaluations, m for the number of low-fidelity model evaluations, and p for an equivalent unit of computational cost that may depend on both n and m.

We also define the operators $\mathbb{E}$ as the expectation of a random variable, Var as the variance of a random variable, Cov as the covariance between two random variables, and Corr as the correlation between two random variables.

Finally, we use bold font to indicate vector quantities when we generalize our results to multiple model outputs and/or multiple statistics of interest. We reserve q for the length of the vector.

1.3 Problem Formulation

Let $M_{\text{high}}(\mathbf{x}, \mathbf{u})$ be the high-fidelity model of an engineering system of interest. We consider the case where there are uncertainties in the input parameters represented by some probability distribution. Therefore, the vector of input parameters $\mathbf{u}$ is a realization of the random vector $\mathbf{U}(\omega)$ and the model output is a random variable $A(\mathbf{x}, \omega) = M_{\text{high}}(\mathbf{x}, \mathbf{U}(\omega))$. In this setting, the statistic of the model output $s_A(\mathbf{x})$, such as the mean and the variance, can be used to describe the system performance. In most cases and especially for complex engineering models, we cannot determine $s_A(\mathbf{x})$ exactly and so we develop an estimator $\hat{s}_A(\mathbf{x})$ that can be computed efficiently. When there are multiple model outputs and/or multiple statistics of interest, we generalize to the vector of estimators $\hat{\mathbf{s}}_A(\mathbf{x})$ for the vector of statistics $\mathbf{s}_A(\mathbf{x})$.

To perform design under uncertainty, we optimize the statistics $\mathbf{s}_A(\mathbf{x})$. Therefore, we consider the following general optimization problem:

$$\begin{aligned} \mathbf{x}^* = \arg\min_{\mathbf{x}} \quad & f(\mathbf{x}, \mathbf{s}_A(\mathbf{x})) \\ \text{s.t.} \quad & g(\mathbf{x}, \mathbf{s}_A(\mathbf{x})) \leq 0 \\ & h(\mathbf{x}, \mathbf{s}_A(\mathbf{x})) = 0, \end{aligned} \tag{1.1}$$

where the objective and constraint functions f, g, and h may depend on the statistics $\mathbf{s}_A(\mathbf{x})$. For example, in robust design, the objective function may be the mean plus a constant factor of the standard deviation of the model output. Since $\mathbf{s}_A(\mathbf{x})$ is typically unknown, we approximate it with its estimator $\hat{\mathbf{s}}_A(\mathbf{x})$ and so the objective and constraints are themselves also estimators: $\hat{f}(\mathbf{x}) = f(\mathbf{x}, \hat{\mathbf{s}}_A(\mathbf{x}))$, $\hat{g}(\mathbf{x}) = g(\mathbf{x}, \hat{\mathbf{s}}_A(\mathbf{x}))$, and $\hat{h}(\mathbf{x}) = h(\mathbf{x}, \hat{\mathbf{s}}_A(\mathbf{x}))$.

This thesis focuses on reducing the computational cost of computing the estimator $\hat{s}_A(\mathbf{x})$ by making use of approximate information in a rigorous manner. We assume that the computational cost is dominated by the repeated evaluation of the high-fidelity model—the algebraic cost of the algorithms is small relative to the cost of the engineering models considered. In Chapter 2, we make use of a low-fidelity model $M_{\text{low}}(\mathbf{x}, \mathbf{u})$ that is inexpensive to evaluate relative to the high-fidelity model $M_{\text{high}}(\mathbf{x}, \mathbf{u})$. The low-fidelity model takes the same vector of design variables $\mathbf{x}$ and the same realization of the vector of input parameters $\mathbf{u}$ but returns a different output. In Chapter 3, we take advantage of the autocorrelation of the high-fidelity model over the design space. In both cases, the estimator $\hat{s}_A(\mathbf{x})$ converges to the true statistics $s_A(\mathbf{x})$ regardless of the quality of the approximate information; however, a good and inexpensive approximation[1] will provide the greatest benefit in reducing computational cost.

1.4 Literature Review

Given the background, we now review the literature on related topics in multifidelity optimization and uncertainty quantification. We also review the control variate method, which serves as the framework for the developments in Chapter 2 and Chapter 3, as well as the related multilevel Monte Carlo method.

1.4.1 Multifidelity Methods in Optimization

Some of the pioneering work on multifidelity methods is in deterministic optimization to reduce the computational cost of finding the optimum of an expensive high-fidelity model. There are two important classes of multifidelity optimization methods: trust-region-based methods and pattern-search-based methods. In a trust-region method, a surrogate is constructed from the high-fidelity model evaluations and is minimized within a trust-region to generate candidate steps in the design space. The size of the trust-region is adjusted based on the predictive capability of the surrogate. The Approximation and Model Management Optimization (AMMO) [1] extends the general trust-region framework to multifidelity models by defining the surrogate as the low-fidelity model and a correction. Convergence to the high-fidelity model local optimum is guaranteed by ensuring that the correction makes the surrogate have the same value and gradient as the high-fidelity model at the center of the trust-region

[1] We define what constitutes a "good" approximation in Chapter 2 and Chapter 3.

[6]. A modification to the general framework removes the need to evaluate the high-fidelity model gradients, which may be unavailable and difficult to estimate in some situations [32, 33].

In a pattern search method, the high-fidelity model is evaluated without gradients on a stencil in the design space called a poll step to try to find an improved design point. The stencil is constructed on an underlying mesh whose fineness is adjusted based on the success of the poll step [49]. Additional flexibility is provided by an optional search step that precedes the poll step to find an improved design point on the mesh by any means available. A successful search step allows the poll step to be skipped. The surrogate management framework (SMF) [4] takes advantage of this flexibility by using the inexpensive low-fidelity model in the search step to try to find an improved design point and skip the expensive poll step. Furthermore, if the poll step is required, the order of high-fidelity model evaluations on the stencil can be arranged based on the values of the low-fidelity model evaluations on the same stencil to try to find an improved design point earlier and skip the rest of the poll step. Thus, in the surrogate management framework, the high-fidelity model serves to check the suggestions provided by the low-fidelity model.

The work on multifidelity optimization developed some of the key ideas for effective general multifidelity approaches that we also apply to our approach: perform the computation on the inexpensive low-fidelity model and evaluate the expensive high-fidelity model as needed only as a check or a correction. A shortcoming in many of these methods is that they do not take into account the relative computational cost between the high-fidelity model and the low-fidelity model and often assume that the low-fidelity model can be evaluated in negligible time compared to the evaluation of the high-fidelity model. When this is not true, the large number of low-fidelity model evaluations can become a significant computational burden. A corollary of this is that these multifidelity methods cannot, in general, guarantee that they reduce the computational cost relative to the single fidelity method, although numerical experiments in the above work suggest that they are often effective in practice.

1.4.2 Multifidelity Methods in Uncertainty Quantification

An early work to bring multifidelity ideas to uncertainty quantification is to use Bayesian regression to construct a Gaussian process emulator of the high-fidelity model using both high-fidelity model evaluations and low-fidelity model evaluations, where the high-fidelity model evaluations are a subset of the low-fidelity model evaluations [25]. The structure of the emulator is based on the assumption that, given the evaluation of the low-fidelity model at a particular design point, we cannot learn more about the high-fidelity model at that design point from an evaluation of the low-fidelity model at another design point. The emulator can then be used for uncertainty quantification, such as estimating the statistics of the high-fidelity model. As an alternative to a Gaussian process emulator of the high-fidelity model, a stochastic expansion, such as polynomial chaos or stochastic collocation, can be constructed based on a sparse grid, where the high-fidelity model sparse grid is a subset of the low-fidelity model sparse grid [37]. The high-fidelity model evaluations are used to generate a stochastic expansion of the correction between the low-fidelity model and the high-fidelity model and is then combined with the stochastic expansion of the low-fidelity model. This approach is effective if the correction can be approximated accurately with a stochastic expansion of a lower order than that of the high-fidelity model.

In rare event simulation (e.g., failure probability in reliability analysis), the computational expense is due to the low probability of event occurrence and the need to simulate the high-fidelity model near the event boundary. When multifidelity models are available, a large number of inexpensive low-fidelity model simulations can be performed to narrow down the location of the event boundary so that fewer expensive high-fidelity model simulations are needed to refine the event boundary [11, 30]. A multifidelity approach can be particularly effective for rare event simulation because the values of the high-fidelity model outputs are needed only in a small region of the stochastic space. This is similar to optimization where the values of the high-fidelity model are needed only near the path in the design space towards the optimum. In both of these situations, we are interested in the local behaviour of the high-fidelity model. This setting enables the use of the inexpensive low-fidelity model to search for the region of interest in order to focus the expensive high-fidelity model on that region.

Unfortunately, other statistics, such as the mean and the variance, are global in the sense that they require high-fidelity model outputs over the entire stochastic space. Therefore, the methods for rare event simulation are inappropriate and it can be challenging to extract the benefits of the low-fidelity model with methods that construct an emulator of the high-fidelity model over the entire stochastic space. In this thesis, we investigate an alternative approach that performs a correction to the low-fidelity model at the level of the aggregate statistics rather than at the level of the model outputs. Central to this idea is the method of control variates that allows one to compute statistics of a random variable with the help of an auxiliary random variable.

1.4.3 Control Variate Methods

In Monte Carlo (MC) simulation, we generate samples of the numerical model outputs from the distributions of the random input parameters to estimate statistics of interest, such as the means and the variances of the model outputs. Due to the randomness of the samples, the estimators are themselves also random and their accuracy can be measured in terms of their mean squared errors (MSE) or root mean squared errors (RMSE). For an unbiased estimator, the MSE is simply the estimator variance. In regular Monte Carlo simulation, the convergence rate of the RMSE of an estimator scales as $n^{-1/2}$, where n is the number of samples. The convergence rate is independent of the number of random inputs, making Monte Carlo simulation suitable for problems with a large number of input parameters. It is also simple to implement, using only repeated evaluations of the numerical model, which can be provided as a closed "black-box". However, because of the slow rate of convergence, it may be necessary to generate many samples of the model outputs to achieve the desired accuracy, requiring many evaluations of the numerical model.

A variety of techniques have been developed to reduce the estimator variance in Monte Carlo simulation, such as antithetic variates, control variates, importance sampling, conditional Monte Carlo sampling, and stratified sampling [18]. These methods do not increase the rate of convergence, but they reduce the constant factor in the RMSE. In particular, the control variate (CV) method reduces the estimator variance of a statistic of the random variable $A(\omega)$ by making use of the correlation between $A(\omega)$ and an auxiliary random variable $B(\omega)$ [35]. We based our multifidelity approach on this framework by defining $A(\omega)$ to be the high-fidelity model output and $B(\omega)$ to be the low-fidelity model output, where the randomness of both model outputs are induced by the same random inputs. In the classical control variate method, the exact value of the statistic of the auxiliary random variable $B(\omega)$ is required. Extensions to the control variate method that relax this requirement include the biased control variate (BCV) method [44], which replaces the exact statistic of $B(\omega)$ with a deterministic approximation, and the quasi control variate (QCV) method [14] and the control variates using estimated means (CVEM) method [40], both of which replace the exact statistic of $B(\omega)$ with an estimate from a prior simulation. In contrast to QCV and CVEM, our multifidelity approach does not utilize a prior (or offline) simulation. Instead, we generate a single stream of the random inputs and evaluate the low-fidelity model on this stream of inputs to generate samples of $B(\omega)$ and evaluate the high-fidelity model on a subset of this stream of inputs to generate samples of $A(\omega)$. We make use of the relative computational cost between the high-fidelity model and the low-fidelity model to balance the number of samples of $A(\omega)$ and the number of samples of $B(\omega)$ generated.

The StackedMC method [50] is another extension of the classical control variate method that is related to our multifidelity approach. Instead of a provided low-fidelity model, the StackedMC method constructs a surrogate from the samples of the high-fidelity model using supervised learning techniques. This surrogate can then be used as the auxiliary random variable $B(\omega)$ in the control variate method. The StackedMC method is useful when a low-fidelity model is not readily available. Alternatively, it may be combined with our multifidelity approach to further improve efficiency.

In the context of optimization under certainty, such as problem (1.1), it may be necessary to estimate the statistics of the high-fidelity model at many different values of the design variables. In this setting, the high-fidelity model output at each vector of design variables can be thought of as a different random variable, providing a selection of candidates for the auxiliary random variable in the control variate method. The databased Monte Carlo (DBMC) method [5] is also developed for the case where it is necessary to estimate the statistics at many different values of some deterministic input parameters. DBMC stores the simulation result at a particular set of the deterministic input parameters in a database and use it in the control variate method to improve the estimate of the statistics at other sets of the deterministic input parameters. In contrast, we develop an approach tailored to optimization under uncertainty, where the deterministic input parameters are the design variables provided by an external optimization algorithm and the "database" is updated with each new vector of design variables.

1.4.4 Multilevel Monte Carlo Methods

While the control variate method can be effective in reducing the estimator variance, the convergence rate remains at $n^{-1/2}$, where n is the number of samples, representing the computational cost of the simulation. The multilevel Monte Carlo (MLMC) method [15, 48] increases the rate of convergence by making use of multiple levels of lower fidelity models analogous to multigrid methods in the solution of partial differential equations. Let the high-fidelity model be a solver for a stochastic differential equation (SDE) with the desired step size. A sequence of less expensive and lower fidelity models can be generated by increasingly coarsening the step size, which is then used as multiple control variates to estimate the statistics of the SDE solution at the original step size. Since the computational cost of the SDE solver is inversely proportional to the step size, MLMC can achieve an increased convergence rate as a function of the total computational cost by using a geometric sequence of increasing step sizes.

Our multifidelity approach is related to the MLMC method in that a low-fidelity model is used in the control variate method to reduce the error of the statistic estimator. Thus, our multifidelity approach may be interpreted as the MLMC method with one level, although the distinguishing MLMC feature of increased convergence rate is lost if only one level is used. Since we do not use a sequence of lower fidelity models with geometrically decreasing computational costs and model errors, we can accept more general types of multifidelity models, as described at the end of §1.1.

The multilevel idea has also been applied to the case where it is necessary to estimate the statistics of the high-fidelity model at many different values of some deterministic input parameters. In some applications, it may be desirable to estimate the function of the statistics in terms of the deterministic input parameters. This may be done by interpolating the estimators of the statistics over a grid of the deterministic input parameters. A multilevel approach can be applied here by successively coarsening the grid of the deterministic input parameters at each level [19]. In the case of optimization under uncertainty, the deterministic input parameters are the design variables. However, the design variables are provided by an external optimization algorithm to search for the optimum and are not in the form of a grid with spacing that can be controlled in a multilevel framework. Furthermore, it is not necessary to obtain a functional form of the statistic estimator over the entire design space. Instead, we make use of estimators only at the design variables chosen by the optimization algorithm to step toward the optimum.

1.5 Thesis Objectives

Based on our motivation and the literature review, we find a need for a general multifidelity method for uncertainty propagation that can be applied to a wide range of models, especially those representing large-scale engineering systems. To further improve efficiency, we need to exploit the structure provided by the optimization under uncertainty setting to extract additional approximate information. To summarize, the high level objectives of this thesis are:

- to develop rigorous approaches that leverage inexpensive surrogate models and approximate information to reduce the computational cost of optimization under uncertainty; and

- to demonstrate the effectiveness of the multifidelity approaches in practical aircraft design under uncertainty problems.

1.6 Thesis Outline

In Chapter 2, we develop the multifidelity estimator of the statistics of the high-fidelity model based on the control variate method. We show that the multifidelity estimator reduces computational cost by automatically balancing the computational load between high-fidelity model evaluations and low-fidelity model evaluations. In Chapter 3, we develop the information reuse estimator that utilizes estimators at previous optimization iterations in the control variate method to reduce computational cost. This estimator is used specifically in an optimization under uncertainty setting and is increasingly efficient as the optimizer approaches the optimal solution. We also show how the multifidelity estimator and the information reuse estimator can be combined to further improve efficiency. In Chapter 4, we demonstrate the effectiveness of our methods in practical aerospace applications. Finally, we summarize the thesis contributions in Chapter 5.

Chapter 2

Multifidelity Models in Monte Carlo Simulation

In this chapter, we develop a method to compute the estimator $\hat{s}_A$ of the exact statistic s_A, where the random variable $A(\omega)$ is the high-fidelity model output $M_{\text{high}}(\mathbf{x}, \mathbf{U}(\omega))$ at some fixed values of the design variables $\mathbf{x}$. In §2.1, we introduce the multifidelity estimator that makes use of the random output of the low-fidelity model $M_{\text{low}}(\mathbf{x}, \mathbf{U}(\omega))$. Next, we discuss implementation issues in §2.2. In §2.3, we return to the optimization problem (1.1) and discuss several optimization algorithms to use in conjunction with the multifidelity estimator. Finally, we demonstrate the effectiveness of the multifidelity estimator with numerical examples in §2.4.

2.1 Approach

Given an arbitrary random variable $A(\omega)$, consider the problem of estimating $s_A = \mathbb{E}\left[A(\omega)\right]$. Given n independent and identically distributed (i.i.d.) samples $a_1, a_2, \ldots, a_n$ drawn from the distribution of $A(\omega)$, the regular Monte Carlo estimator of s_A, denoted as $\bar{a}_n$, is

$$\bar{a}_n = \frac{1}{n}\sum_{i=1}^{n} a_i$$

and its mean square error (MSE) is given by the estimator variance

$$\mathrm{MSE}\left[\bar{a}_n\right] = \mathrm{Var}\left[\bar{a}_n\right] = \frac{1}{n^2}\,\mathrm{Var}\left[\sum_{i=1}^{n} a_i\right] = \frac{\sigma_A^2}{n}, \tag{2.1}$$

where $\sigma_A^2 = \mathrm{Var}\left[A\right]$ is the (unknown) variance of $A(\omega)$. To reduce the number of samples needed to achieve an acceptably low estimator variance, we focus on the control variate method [18, 35], which makes use of the correlation between the random variable $A(\omega)$ and an auxiliary random variable $B(\omega)$. In the multifidelity setting, we take $A(\omega)$ to be the random output of the high-fidelity model and $B(\omega)$ to be the random output of the low-fidelity model. Therefore, we first review the control variate method.

2.1.1 Control Variate Estimator

Again, we consider estimating $s_A = \mathbb{E}\left[A(\omega)\right]$. Given an auxiliary random variable $B(\omega)$ in which the statistic s_B is known exactly (in this case, $s_B = \mathbb{E}\left[B(\omega)\right]$), let $\{a_i, b_i\}_{i=1}^n$ be n i.i.d. pairs of samples drawn from the joint distribution of $A(\omega)$ and $B(\omega)$. The control variate estimator of s_A, denoted as $\hat{s}_A$, is

$$\hat{s}_A = \bar{a}_n + \alpha\big(s_B - \bar{b}_n\big)$$

for some control parameter $\alpha \in \mathbb{R}$, where

$$\bar{a}_n = \frac{1}{n}\sum_{i=1}^{n} a_i, \qquad \bar{b}_n = \frac{1}{n}\sum_{i=1}^{n} b_i.$$

The control variate formulation can be interpreted as an adjustment to the regular Monte Carlo estimator $\bar{a}_n$. Since $\mathbb{E}\left[s_B - \bar{b}_n\right] = 0$, the second term has no effect in expectation. However, for finite sample size n, $\alpha\big(s_B - \bar{b}_n\big)$ represents an adjustment to $\bar{a}_n$ based on the error of $\bar{b}_n$ with respect to the exact statistic s_B.

The variance of the control variate estimator is

$$\begin{aligned}
\operatorname{Var}\left[\hat{s}_A\right] &= \operatorname{Var}\left[\bar{a}_n\right] + \alpha^2 \operatorname{Var}\left[\bar{b}_n\right] - 2\alpha \operatorname{Cov}\left[\bar{a}_n, \bar{b}_n\right] \\
&= \frac{\sigma_A^2}{n} + \alpha^2 \frac{\sigma_B^2}{n} - 2\alpha \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \operatorname{Cov}\left[a_i, b_j\right] \\
&= \frac{\sigma_A^2}{n} + \alpha^2 \frac{\sigma_B^2}{n} - 2\alpha \frac{1}{n^2} \sum_{i=1}^{n} \operatorname{Cov}\left[a_i, b_i\right] \\
&= \frac{1}{n}\left(\sigma_A^2 + \alpha^2 \sigma_B^2 - 2\alpha \rho_{AB} \sigma_A \sigma_B\right),
\end{aligned}$$

where σ_A^2 is the (unknown) variance of $A(\omega)$, σ_B^2 is the (unknown) variance of $B(\omega)$, and ρ_{AB} is the (unknown) correlation coefficient between $A(\omega)$ and $B(\omega)$.[1] Minimizing $\operatorname{Var}\left[\hat{s}_A\right]$ with respect to α results in

$$\alpha^* = \rho_{AB} \frac{\sigma_A}{\sigma_B}$$

and

$$\operatorname{MSE}\left[\hat{s}_A^*\right] = \operatorname{Var}\left[\hat{s}_A^*\right] = \left(1 - \rho_{AB}^2\right) \frac{\sigma_A^2}{n}.$$

Thus, a reduction in the variance of the estimator can be achieved if $B(\omega)$ is correlated with $A(\omega)$—the higher the correlation, the greater the reduction in estimator variance for a fixed number of samples n. For the ideal case of $\rho_{AB} = 1$, $\operatorname{Var}\left[\hat{s}_A^*\right] = 0$ for any number samples.

The control variate method can be interpreted from a regression point of view. Figure 2-1 illustrates a scatter plot of n samples of $A(\omega)$ and $B(\omega)$ and the linear regression of the samples with slope α. The locations of $\bar{a}_n$ and $\bar{b}_n$ are computed from these samples. Given s_B, the control variate method makes an adjustment to $\bar{a}_n$ from the error in $\bar{b}_n$ with respect to s_B and the slope of the regression line α to obtain $\hat{s}_A$. Based on this framework, we now present the multifidelity estimator.

[1] We discuss the estimation of these parameters in §2.2.1.

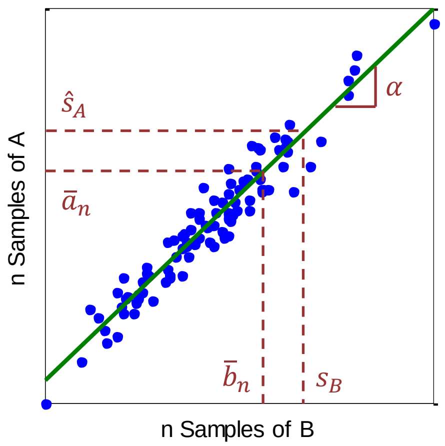


Figure 2-1: Illustration of the regression interpretation of the control variate method. The dots are samples of the random variables $A(\omega)$ and $B(\omega)$.

2.1.2 Multifidelity Estimator

We consider the estimation of the mean of a high-fidelity model at a fixed vector of design variables. Thus, we drop $\mathbf{x}$ from the notation for clarity. Let the random variable $A(\omega) = M_{\text{high}}(\mathbf{U}(\omega))$ be the random output of the high-fidelity model and $B(\omega) = M_{\text{low}}(\mathbf{U}(\omega))$ be the random output of the low-fidelity model. As we shall see, for the low-fidelity model to be useful, it should be cheap to evaluate and $B(\omega)$ should be correlated with $A(\omega)$.

We adapt the control variate method to multifidelity models using the samples of model outputs $a_i = M_{\text{high}}(\mathbf{u}_i)$ and $b_i = M_{\text{low}}(\mathbf{u}_i)$, where $\mathbf{u}_i$, $i = 1, 2, 3, \ldots$ are i.i.d. samples drawn from the distribution of the random input vector $\mathbf{U}(\omega)$. However, $s_B = \mathbb{E}\left[M_{\text{low}}(\mathbf{U}(\omega))\right]$ is not known in this case and is approximated by $\bar{b}_m = \frac{1}{m}\sum_{i=1}^{m} b_i$ with $m \gg n$. Some other extensions to the control variate method address this problem by generating an independent simulation with m samples to compute $\bar{b}_m$ [14, 40]. In our case, we simply require $m - n$ additional samples of b_i beyond the n samples already available. Thus, the multifidelity estimator of s_A, denoted as $\hat{s}_{A,p}$, is

$$\hat{s}_{A,p} = \bar{a}_n + \alpha\left(\bar{b}_m - \bar{b}_n\right) = \alpha\bar{b}_m + \left(\bar{a}_n - \alpha\bar{b}_n\right) \tag{2.2}$$

for some control parameter $\alpha \in \mathbb{R}$ and computational effort p to be defined below.

The second equality emphasizes the interpretation that the majority of the model evaluations is performed on the low-fidelity model and then a correction term is applied. The variance of the multifidelity estimator is

$$\begin{aligned}
\operatorname{Var}\left[\hat{s}_{A,p}\right] = &\operatorname{Var}\left[\bar{a}_n\right] + \alpha^2 \operatorname{Var}\left[\bar{b}_m\right] + \alpha^2 \operatorname{Var}\left[\bar{b}_n\right] \\
&+ 2\alpha \operatorname{Cov}\left[\bar{a}_n, \bar{b}_m\right] - 2\alpha \operatorname{Cov}\left[\bar{a}_n, \bar{b}_n\right] - 2\alpha^2 \operatorname{Cov}\left[\bar{b}_m, \bar{b}_n\right] \\
=& \frac{\sigma_A^2}{n} + \alpha^2 \frac{\sigma_B^2}{m} + \alpha^2 \frac{\sigma_B^2}{n} \\
&+ 2\alpha \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \operatorname{Cov}\left[a_i, b_j\right] - 2\alpha \frac{\rho_{AB}\sigma_A\sigma_B}{n} - 2\alpha^2 \frac{1}{nm} \sum_{i=1}^{m} \sum_{j=1}^{n} \operatorname{Cov}\left[b_i, b_j\right] \\
=& \frac{\sigma_A^2}{n} + \alpha^2 \frac{\sigma_B^2}{m} + \alpha^2 \frac{\sigma_B^2}{n} \\
&+ 2\alpha \frac{1}{nm} \sum_{i=1}^{n} \operatorname{Cov}\left[a_i, b_i\right] - 2\alpha \frac{\rho_{AB}\sigma_A\sigma_B}{n} - 2\alpha^2 \frac{1}{nm} \sum_{j=1}^{n} \operatorname{Cov}\left[b_j, b_j\right] \\
=& \frac{1}{n}\left(\sigma_A^2 + \alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B\right) - \frac{1}{m}\left(\alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B\right).
\end{aligned}$$

The computational effort to compute the regular Monte Carlo estimator from (2.1) is n evaluations of the high-fidelity model. However, computing the multifidelity estimator $\hat{s}_{A,p}$ requires n evaluations of the high-fidelity model and m evaluations of the low-fidelity model. To benchmark against the regular Monte Carlo estimator, we define a unit of computational effort p based on the "equivalent number of high-fidelity model evaluations":

$$p = n + \frac{m}{w} = n\left(1 + \frac{r}{w}\right),$$

where w is the (assumed known) ratio of the average computation time per high-fidelity model evaluation to the average computation time per low-fidelity model evaluation and $r = m/n > 1$ is the ratio of the number of low-fidelity model evaluations to the number of high-fidelity model evaluations. Therefore, in addition to determining the optimal α, we must allocate the computational budget between high-fidelity model evaluations and low-fidelity model evaluations. We rewrite the multifidelity estimator variance in terms of p, r, and w as

$$\mathrm{Var}\left[\hat{s}_{A,p}\right] = \underbrace{\frac{1}{p}\Big(1+\frac{r}{w}\Big)}_{\frac{1}{n}}\left[\sigma_A^2 + \Big(1-\frac{1}{r}\Big)\big(\alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B\big)\right].$$

Given a fixed computational budget p, we minimize $\mathrm{Var}\left[\hat{s}_{A,p}\right]$ in terms of both α and r. The result is

$$\alpha^* = \rho_{AB}\frac{\sigma_A}{\sigma_B}, \qquad r^* = \sqrt{\frac{w\rho_{AB}^2}{1-\rho_{AB}^2}}$$

and

$$\mathrm{MSE}\left[\hat{s}_{A,p}^*\right] = \mathrm{Var}\left[\hat{s}_{A,p}^*\right] = \Big(1+\frac{r^*}{w}\Big)\left[1-\Big(1-\frac{1}{r^*}\Big)\rho_{AB}^2\right]\frac{\sigma_A^2}{p}. \tag{2.3}$$

It can be seen that r^* allocates a greater proportion of the computational budget p to low-fidelity model evaluations when the low-fidelity model is inexpensive (i.e., w is large) and "accurate", where accuracy is in terms of the correlation (i.e., ρ_{AB} is close to 1). Nevertheless, the penalty for approximating the exact statistic s_B with $\bar{b}_m$ is that we achieve less variance reduction than the original control variate method.

If $w \to \infty$, i.e., the low-fidelity model is almost free, then $r^* \to \infty$ and $\mathrm{Var}\left[\hat{s}_{A,p}^*\right] \to (1-\rho_{AB}^2)\frac{\sigma_A^2}{p}$ so that we recover the classical control variate solution. On the other hand, if $\rho_{AB} \to 1$, i.e., the low-fidelity model is almost perfect, then $\mathrm{Var}\left[\hat{s}_{A,p}^*\right] \to \frac{1}{w}\frac{\sigma_A^2}{p}$. Therefore, a perfectly correlated low-fidelity model is not sufficient for variance reduction over the regular Monte Carlo estimator using the same computational budget p; it must also be cheaper to evaluate than the high-fidelity model, i.e. $w > 1$. However, if the values of w and ρ_{AB} are such that $r^* \leq 1$, the correlation is not high enough and the low-fidelity model is not cheap enough for the multifidelity estimator to be worthwhile. In other words, given a low-fidelity model that is w times cheaper to evaluate than the high-fidelity model, it is useful only if its correlation coefficient with respect to the high-fidelity model satisfies

$$\rho_{AB}^2 > \frac{1}{1+w}.$$

If this is not the case, it is better to switch back to the regular Monte Carlo estimator.

2.2 Implementation

The key pieces for the implementation of the multifidelity Monte Carlo simulation are the calculation of the parameters α^* and r^* and an iterative procedure to increment the number of samples. We also discuss practical aspects including estimating statistics other than the mean and estimating functions of statistics.

2.2.1 Parameter Calculations

In practice, since σ_A, σ_B, and ρ_{AB} are unknown, the optimal parameters α^* and r^* are replaced by their sample estimates $\hat{\alpha}$ and $\hat{r}$ based on the n samples of $A(\omega)$ and $B(\omega)$, $\{a_i, b_i\}_{i=1}^n$:

$$\hat{\alpha} = \frac{\sum_{i=1}^n (a_i - \bar{a}_n)\big(b_i - \bar{b}_n\big)}{\sum_{i=1}^n \big(b_i - \bar{b}_n\big)^2}, \tag{2.4a}$$

$$\hat{r} = \sqrt{\frac{w\hat{\rho}_{AB}^2}{1 - \hat{\rho}_{AB}^2}}, \tag{2.4b}$$

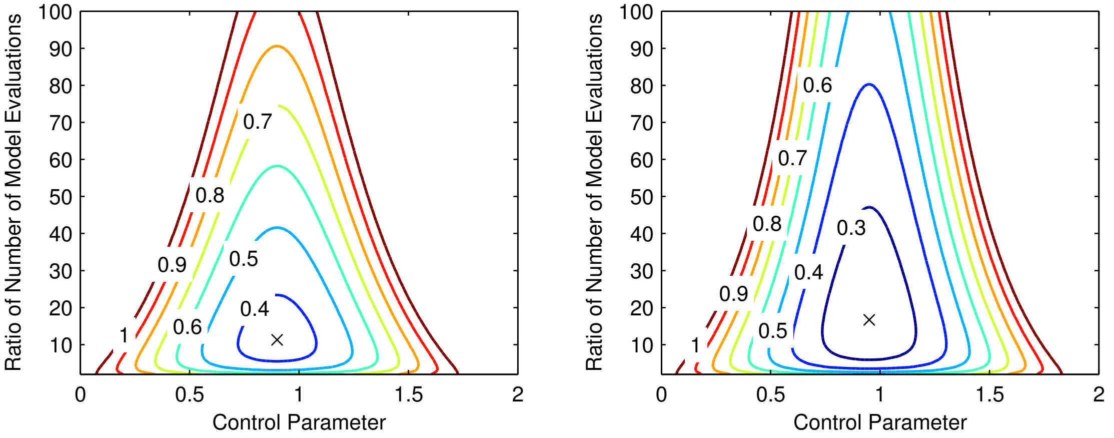
$$\hat{\rho}_{AB}^2 = \frac{\left[\sum_{i=1}^n (a_i - \bar{a}_n)\big(b_i - \bar{b}_n\big)\right]^2}{\left[\sum_{i=1}^n (a_i - \bar{a}_n)^2\right]\left[\sum_{i=1}^n \big(b_i - \bar{b}_n\big)^2\right]}, \tag{2.4c}$$

$$\hat{\sigma}_A^2 = \frac{\sum_{i=1}^n (a_i - \bar{a}_n)^2}{n-1}. \tag{2.4d}$$

To assess the impact of the errors in $\hat{\alpha}$ and $\hat{r}$ relative to the exact parameters α^* and r^*, we plot the ratio

$$\frac{\operatorname{Var}\left[\hat{s}_{A,p}\right]}{\operatorname{Var}\left[\bar{a}_p\right]} = \left(1 + \frac{r}{w}\right)\left[1 + \left(1 - \frac{1}{r}\right)\left(\alpha^2\frac{\sigma_B^2}{\sigma_A^2} - 2\alpha\rho_{AB}\frac{\sigma_B}{\sigma_A}\right)\right]$$

as a function of α and r in Figure 2-2 for some typical values of w and ρ_{AB}. A value less than one indicates a reduction in variance over the regular Monte Carlo estimator at the same computational effort. It can be seen that there is reasonable room for deviations from α^* and r^* (location indicated by the cross), although optimal variance reduction will not be achieved.



(a) Correlation Coefficient $\rho_{AB} = 0.9$ (b) Correlation Coefficient $\rho_{AB} = 0.95$

Figure 2-2: Contour plots of $\operatorname{Var}[\hat{s}_{A,p}] / \operatorname{Var}[\bar{a}_p]$ as a function of control parameter α and ratio of the number of model evaluations r with $w = 30$ and $\sigma_A/\sigma_B = 1$. The cross indicates the location of (α^*, r^*).

2.2.2 Iterative Procedure

The procedure to compute the multifidelity estimator begins with a set of $n = n_{\text{init}}$ samples $\{a_i, b_i\}_{i=1}^n$ that is incremented by n_Δ every iteration. The computation times for the initial n_{init} samples can be used to determine w, if not already available, and n_Δ may be chosen based on the number of processors available for parallel evaluations. During each iteration, the samples $\{a_i, b_i\}_{i=1}^n$ are used to calculate the parameters $\hat{\alpha}$ and $\hat{r}$. This then determines the total number of low-fidelity model evaluations needed to compute the multifidelity estimator (including the existing n low-fidelity model evaluations) as $m = n\hat{r}$. Omitting algorithm overhead, the computational expense is therefore $p = n + m/w$. An algorithm to compute the multifidelity estimator $\hat{s}_{A,p}$ for $s_A = \mathbb{E}[A(\omega)] = \mathbb{E}[M_{\text{high}}(\mathbf{U}(\omega))]$ is shown in Algorithm 2.1. We emphasize that all of the samples used in the algorithm are generated from the same stream of random input vectors $\mathbf{u}_i$, $i = 1, 2, 3, \ldots$ in order to induce the correlation needed for the multifidelity estimator.

Algorithm 2.1 Multifidelity Estimator

```
Given desired RMSE, ratio of average computation time $w$, initial number of samples $n_{\text{init}}$, increment in number of samples $n_\Delta$, high-fidelity model $M_{\text{high}}(\mathbf{u})$, low-fidelity model $M_{\text{low}}(\mathbf{u})$, and a sequence of pseudo-random input vectors $\mathbf{u}_i$ for $i = 1, 2, 3, \ldots$ drawn from the distribution of $\mathbf{U}(\omega)$:
1  Let $n_{\text{old}} = 0$, $m_{\text{old}} = 0$, $\ell = 0$, and $n = n_{\text{init}}$.
2  Evaluate samples $a_i = M_{\text{high}}(\mathbf{u}_i)$ for $i = n_{\text{old}} + 1, \ldots, n$.
3  If $\ell < n$, evaluate samples $b_i = M_{\text{low}}(\mathbf{u}_i)$ for $i = \ell + 1, \ldots, n$ and set $\ell = n$.
4  Compute $\bar{a}_n$ using $\{a_i\}_{i=1}^n$ and $\bar{b}_n$ using $\{b_i\}_{i=1}^n$.
5  Compute $\hat{\alpha}$, $\hat{\rho}_{AB}^2$, $\hat{r}$ and $\hat{\sigma}_A^2$, from (2.4) using $\{a_i, b_i\}_{i=1}^n$.
6  Set $m \leftarrow \max\{n\hat{r}, m\}$.
7  If $\ell < m$, evaluate samples $b_i = M_{\text{low}}(\mathbf{u}_i)$ for $i = \ell + 1, \ldots, m$ and set $\ell = m$.
8  Compute $\bar{b}_m$ using using $\{b_i\}_{i=1}^m$
9  Compute multifidelity estimator from (2.2).
10 Compute RMSE from (2.3).
11 If RMSE is too large, set $n_{\text{old}} = n$, $m_{\text{old}} = m$, $n \leftarrow n + n_\Delta$ and return to Step 2;
   otherwise, stop.
```

2.2.3 Estimating Variance

The preceding development assumes that we are interested in estimating $s_A = \mathbb{E}\left[A(\omega)\right] = \mathbb{E}\left[M_{\text{high}}(\mathbf{U}(\omega))\right]$. We may also use the procedure to estimate other statistics such as $s_A = \operatorname{Var}\left[A(\omega)\right] = \operatorname{Var}\left[M_{\text{high}}(\mathbf{U}(\omega))\right]$, but some care is needed to ensure that the method is efficient. For example, computing the variance based on the formula $\mathbb{E}\left[A(\omega)^2\right] - \left(\mathbb{E}\left[A(\omega)\right]\right)^2$ can produce unsatisfactory results because $\rho_{A^2B^2}$ tends to be worse than ρ_{AB}. Instead, it is usually better to define the samples from the residuals as

$$a_i = \frac{n}{n-1}\left(M_{\text{high}}(\mathbf{u}_i) - \frac{1}{n}\sum_{j=1}^{n} M_{\text{high}}(\mathbf{u}_j)\right)\left(M_{\text{high}}(\mathbf{u}_i) - \frac{1}{n-1}\sum_{j=1}^{n-1} M_{\text{high}}(\mathbf{u}_j)\right)$$

$$b_i = \frac{m}{m-1}\left(M_{\text{low}}(\mathbf{u}_i) - \frac{1}{m}\sum_{j=1}^{m} M_{\text{low}}(\mathbf{u}_j)\right)\left(M_{\text{low}}(\mathbf{u}_i) - \frac{1}{m-1}\sum_{j=1}^{m-1} M_{\text{low}}(\mathbf{u}_j)\right)$$

based on the one-pass algorithm for computing variance [28]. Using the above definition of the samples a_i and b_i, we get

$$\bar{a}_n = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{1}{n-1}\sum_{i=1}^{n}\left(M_{\text{high}}(\mathbf{u}_i) - \frac{1}{n}\sum_{j=1}^{n} M_{\text{high}}(\mathbf{u}_j)\right)^2$$

$$\bar{b}_m = \frac{1}{m}\sum_{i=1}^{m} b_i = \frac{1}{m-1}\sum_{i=1}^{m}\left(M_{\text{low}}(\mathbf{u}_i) - \frac{1}{m}\sum_{j=1}^{m} M_{\text{low}}(\mathbf{u}_j)\right)^2$$

and so we can apply the approach described in §2.1.2 to the redefined samples $\{a_i, b_i\}$ as if we were estimating the mean. The correlation coefficient based on the residuals is usually higher than that based on the square of the random variables.[2] However, a_i and b_i are no longer i.i.d. samples and the theory of §2.1.2 is not valid. Nevertheless, the results in §2.4 suggest that this approach is still effective in estimating the variance.

We can similarly redefine the samples to estimate other quantities of interest, e.g., indicator functions of the model outputs to estimate probabilities. More advanced control variate techniques are available to estimate quantiles [20]. However, specialized methods may be needed to efficiently estimate rare probabilities and reliability metrics [30], since the control variate approach does not directly address the problem of landing few samples in the rare event region.

2.2.4 Estimating Functions of Statistics

In optimization under uncertainty, we may be concerned with functions of one or more statistics. For example, a robust objective function may be formulated as $\mathbb{E}\left[M_{\text{high}}(\mathbf{U}(\omega))\right] + \sqrt{\text{Var}\left[M_{\text{high}}(\mathbf{U}(\omega))\right]}$. Let $\mathbf{s}_A$ be the $q \times 1$ vector of statistics of interest (e.g., $q = 2$ and $\mathbf{s}_A = \left[\mathbb{E}\left[M_{\text{high}}(\mathbf{U}(\omega))\right]\ \text{Var}\left[M_{\text{high}}(\mathbf{U}(\omega))\right]\right]^\top$) and let $\hat{\mathbf{s}}_{A,p}$ be the $q \times 1$ vector of its estimator. The function may then be written as $f(\mathbf{s}_A)$ and we estimate it as $f(\hat{\mathbf{s}}_{A,p})$.

The error in the function estimator $f(\hat{\mathbf{s}}_{A,p})$ can be approximated using a first-order

[2] For example, consider a low-fidelity model that is simply a constant offset of the high-fidelity model.

Taylor expansion about $\mathbf{s}_A$:

$$f(\hat{\mathbf{s}}_{A,p}) - f(\mathbf{s}_A) \approx \nabla_{\mathbf{s}} f(\hat{\mathbf{s}}_{A,p})^\top (\hat{\mathbf{s}}_{A,p} - \mathbf{s}_A).$$

Squaring both sides and taking the expectation, we obtain an approximation of the mean square error [52]

$$\operatorname{MSE}[f(\hat{\mathbf{s}}_{A,p})] \approx \nabla_{\mathbf{s}} f(\hat{\mathbf{s}}_{A,p})^\top \operatorname{Cov}[\hat{\mathbf{s}}_{A,p}] \nabla_{\mathbf{s}} f(\hat{\mathbf{s}}_{A,p}), \tag{2.5}$$

where $\operatorname{Cov}[\hat{\mathbf{s}}_{A,p}]$ is the $q \times q$ covariance matrix of the vector of estimators $\hat{\mathbf{s}}_{A,p}$. Thus, we need to generalize the scalar multifidelity estimator $\hat{s}_{A,p}$ in (2.2) to the vector case $\hat{\mathbf{s}}_{A,p}$.

Let $\mathbf{A}(\omega)$ and $\mathbf{B}(\omega)$ be $q \times 1$ random vectors and let $\mathbf{\Sigma}_A = \operatorname{Cov}[\mathbf{A}(\omega)]$, $\mathbf{\Sigma}_B = \operatorname{Cov}[\mathbf{B}(\omega)]$, and $\mathbf{\Sigma}_{AB} = \operatorname{Cov}[\mathbf{A}(\omega), \mathbf{B}(\omega)]$ be their $q \times q$ covariance matrices and cross-covariance matrix. Also, let $\boldsymbol{\alpha}$ be a $q \times q$ diagonal matrix whose elements are the q (optimal) control parameters for each of the q components of $\hat{\mathbf{s}}_{A,p}$. Then, the $q \times 1$ vector of multifidelity estimators is

$$\hat{\mathbf{s}}_{A,p} = \bar{\mathbf{a}}_n + \boldsymbol{\alpha}\big(\bar{\mathbf{b}}_m - \bar{\mathbf{b}}_n\big)$$

and the $q \times q$ covariance matrix of the vector of multifidelity estimators is

$$\begin{aligned}
\operatorname{Cov}[\hat{\mathbf{s}}_{A,p}] = \; & \operatorname{Cov}[\bar{\mathbf{a}}_n] + \boldsymbol{\alpha} \operatorname{Cov}\big[\bar{\mathbf{b}}_m\big]\boldsymbol{\alpha}^\top + \boldsymbol{\alpha} \operatorname{Cov}\big[\bar{\mathbf{b}}_n\big]\boldsymbol{\alpha}^\top \\
& + \operatorname{Cov}\big[\bar{\mathbf{a}}_n, \bar{\mathbf{b}}_m\big]\boldsymbol{\alpha}^\top + \boldsymbol{\alpha} \operatorname{Cov}\big[\bar{\mathbf{b}}_m, \bar{\mathbf{a}}_n\big] \\
& - \operatorname{Cov}\big[\bar{\mathbf{a}}_n, \bar{\mathbf{b}}_n\big]\boldsymbol{\alpha}^\top - \boldsymbol{\alpha} \operatorname{Cov}\big[\bar{\mathbf{b}}_n, \bar{\mathbf{a}}_n\big] \\
& - \boldsymbol{\alpha} \operatorname{Cov}\big[\bar{\mathbf{b}}_m, \bar{\mathbf{b}}_n\big]\boldsymbol{\alpha}^\top - \boldsymbol{\alpha} \operatorname{Cov}\big[\bar{\mathbf{b}}_n, \bar{\mathbf{b}}_m\big]\boldsymbol{\alpha}^\top \\
= \; & \frac{\mathbf{\Sigma}_A}{n} + \boldsymbol{\alpha}\frac{\mathbf{\Sigma}_B}{m}\boldsymbol{\alpha}^\top + \boldsymbol{\alpha}\frac{\mathbf{\Sigma}_B}{n}\boldsymbol{\alpha}^\top + \frac{\mathbf{\Sigma}_{AB}}{m}\boldsymbol{\alpha}^\top + \boldsymbol{\alpha}\frac{\mathbf{\Sigma}_{BA}}{m} \\
& - \frac{\mathbf{\Sigma}_{AB}}{n}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\frac{\mathbf{\Sigma}_{BA}}{n} - 2\boldsymbol{\alpha}\frac{\mathbf{\Sigma}_B}{m}\boldsymbol{\alpha}^\top \\
= \; & \frac{1}{n}\Big\{\mathbf{\Sigma}_A + \Big(1 - \frac{n}{m}\Big)\Big(\boldsymbol{\alpha}\mathbf{\Sigma}_B\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\mathbf{\Sigma}_{BA} - (\boldsymbol{\alpha}\mathbf{\Sigma}_{BA})^\top\Big)\Big\}.
\end{aligned}$$

The covariance matrix of the vector of multifidelity estimators can then be used to compute the approximation of the mean square error of the function estimator in (2.5). This error estimate is useful for predicting the number of samples needed to control the objective function noise during optimization.

2.3 Optimization Algorithms and Convergence

We return to optimization problem (1.1) and briefly discuss the choice of optimization algorithms to use in conjunction with the multifidelity estimator. Given the vector of design variables $\mathbf{x}_k$ at optimization iteration k, we apply Algorithm 2.1 to compute $\hat{\mathbf{s}}_{A,p}(\mathbf{x}_k)$ and evaluate the functions $\hat{f}(\mathbf{x}_k) = f(\mathbf{x}_k, \hat{\mathbf{s}}_{A,p}(\mathbf{x}_k))$, $\hat{g}(\mathbf{x}_k) = g(\mathbf{x}_k, \hat{\mathbf{s}}_{A,p}(\mathbf{x}_k))$, and $\hat{h}(\mathbf{x}_k) = h(\mathbf{x}_k, \hat{\mathbf{s}}_{A,p}(\mathbf{x}_k))$. Due to the pseudo-randomness of Monte Carlo sampling, the objective and constraint values returned to the optimizer, $\hat{f}(\mathbf{x}_k)$, $\hat{g}(\mathbf{x}_k)$, and $\hat{h}(\mathbf{x}_k)$, are noisy with respect to the exact objective and constraint values $f(\mathbf{x}_k, \mathbf{s}_A(\mathbf{x}_k))$, $g(\mathbf{x}_k, \mathbf{s}_A(\mathbf{x}_k))$, and $h(\mathbf{x}_k, \mathbf{s}_A(\mathbf{x}_k))$ and the optimization problem becomes a stochastic optimization problem. While the level of noise is controlled by the specified root mean square error (RMSE) tolerance in Algorithm 2.1, it nevertheless poses a challenge for any optimization algorithm that is not noise tolerant. Thus, we consider three classes of optimization algorithms: stochastic approximation, sample average approximation, and derivative-free optimization.

2.3.1 Stochastic Approximation

The stochastic approximation method (also known as the Robbins-Monro method) [47] is designed to find at least a local solution to the unconstrained minimization

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \; f(\mathbf{x}, \mathbf{s}_A(\mathbf{x})),$$

assuming the objective function is bounded from below, using only noisy approximations of the objective. Motivated by the steepest descent method, the algorithm generates a new vector of design variables at optimization iteration k as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \nabla_{\mathbf{x}} \hat{f}(\mathbf{x}_k)$$

starting from an initial vector of design variables $\mathbf{x}_0$. The parameters λ_k, $k = 0, 1, 2, \ldots$ is a prescribed sequence of step lengths and $\nabla_{\mathbf{x}} \hat{f}(\mathbf{x}_k)$ is the gradient estimator. Assuming that $\mathbf{x}^*$ is the unique solution, $\mathbf{x}_k \to \mathbf{x}^*$ as $k \to \infty$ if the following conditions are satisfied [47]:

I. Gain sequence: $\lambda_k > 0$, $\lambda_k \to 0$ as $k \to \infty$, $\sum_{k=0}^{\infty} \lambda_k = \infty$, and $\sum_{k=0}^{\infty} \lambda_k^2 < \infty$.

II. Search direction: For some symmetric, positive definite matrix $\mathbf{H}$ and every $0 < \xi < 1$, $\inf_{\xi < \|\mathbf{x}-\mathbf{x}^*\| < 1/\xi} \left(\mathbf{x} - \mathbf{x}^*\right)^\top \mathbf{H} \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{s}_A(\mathbf{x})) > 0$.

III. Mean-zero noise:[3] $\mathbb{E}\left[\nabla_{\mathbf{x}} \hat{f}(\mathbf{x}) - \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{s}_A(\mathbf{x}))\right] = 0$ for all $\mathbf{x}$ and k.

IV. Growth and variance bounds: $\|\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{s}_A(\mathbf{x}))\|^2 + \mathbb{E}\left[\left\|\nabla_{\mathbf{x}} \hat{f}(\mathbf{x}) - \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{s}_A(\mathbf{x}))\right\|^2\right] \leq \nu\left(1 + \|\mathbf{x}\|^2\right)$ for all $\mathbf{x}$ and k and some $\nu > 0$.

Using the theoretically optimal step lengths $\lambda_k = \frac{\lambda}{k+1}$ for some positive constant λ, the asymptotic rate of convergence is $\mathcal{O}\left[k^{-1/2}\right]$.

The stochastic approximation method requires estimates of the objective gradient. The expectation operator and the gradient operator can be interchanged, for example, $\nabla_{\mathbf{x}} \mathbb{E}\left[M(\mathbf{x}, \mathbf{U}(\omega))\right] = \mathbb{E}\left[\nabla_{\mathbf{x}} M(\mathbf{x}, \mathbf{U}(\omega))\right]$, as long as the function and its gradient are continuous and are bounded above and below [47]. Thus, we can apply Algorithm 2.1 to the gradient output of the high-fidelity model and the gradient output of the low-fidelity model and obtain the multifidelity gradient estimator. If gradient output is unavailable, variations on the stochastic approximation method such as the simultaneous perturbation stochastic approximation method construct the gradient estimator using only noisy function evaluations [46]. The basic method can also be extended to constrained optimization by projecting the vectors of design variables iterates into the feasible region. However, the projection technique is only practical for

[3] This may not be satisfied for some objective functions, but there exists an alternative set of conditions that allow for some bias [47].

simple constraints such as variable bounds and cannot, in general, handle nonlinear (potentially also noisy) constraints.

2.3.2 Sample Average Approximation

In the sample average approximation method (also known as the sample path method) [47], the same realizations $\{\mathbf{u}_i\}_{i=1}^{n}$ of the random input vectors $\mathbf{U}(\omega)$ are used to compute the estimators for all optimization iterations. This effectively turns $\hat{f}(\mathbf{x})$, $\hat{g}(\mathbf{x})$, and $\hat{h}(\mathbf{x})$ into deterministic functions of $\mathbf{x}$, permitting a wide range of deterministic constrained nonlinear programming techniques to solve optimization problem (1.1). However, since $\hat{f}(\mathbf{x})$, $\hat{g}(\mathbf{x})$, and $\hat{h}(\mathbf{x})$ are approximations to $f(\mathbf{x}, \mathbf{s}_A(\mathbf{x}))$, $g(\mathbf{x}, \mathbf{s}_A(\mathbf{x}))$, and $h(\mathbf{x}, \mathbf{s}_A(\mathbf{x}))$, respectively, the solution of the deterministic problem using these n realizations of the random input vectors, denoted as $\mathbf{x}^{(n)}$, does not, in general, coincide with true solution $\mathbf{x}^*$ of (1.1). Nevertheless, as long as the function is bounded above and below, $\mathbf{x}^{(n)} \to \mathbf{x}^*$ as $n \to \infty$ at an asymptotic rate of $\mathcal{O}\left[n^{-1/2}\right]$ [47]. For finite sample size n, a confidence bound on the optimality gap can be computed to assess the quality of the solution [31].

In order to use the multifidelity estimator with the sample average approximation method, n, r, and $m = rn$ must remain fixed for all optimization iterations so that the same samples $\{u_i\}_{i=1}^{m}$ can be used to compute $\bar{\mathbf{b}}_m$ and the same subset of the samples $\{u_i\}_{i=1}^{n}$ can be used to compute $\bar{\mathbf{a}}_n$ and $\bar{\mathbf{b}}_n$. Once n is chosen, a practical choice is to fix the value of r at the optimal r^* for the multifidelity estimator at the first optimization iteration. However, it may no longer be optimal for the multifidelity estimators at subsequent optimization iterations.

2.3.3 Derivative-Free Optimization

If it is not possible to fix the realizations of the random input vector $\mathbf{U}(\omega)$ for all optimization iterations, derivative-free optimization methods may be used to solve the noisy (stochastic) optimization problem. Although derivative-free optimization methods are not designed specifically for stochastic optimization problems, they are typically tolerant to small levels of noise in practice [8]. Examples of derivative-free optimization algorithms include mesh adaptive direct search (MADS) [2], implicit filtering [23], derivative-free optimization (DFO) [7], bound optimization by quadratic approximation (BOBYQA) [42], and constrained optimization by linear approximation (COBYLA) [41]. Most of these methods sample the objective and constraint functions relatively widely in the design space to determine the search direction, which has the effect of smoothing out the high-frequency noise in the function evaluations provided that the magnitude of the noise is small relative to the true function values. In typical practice, this allows the optimizer to find a solution that is close to the true optimum $\mathbf{x}^*$ of (1.1). However, there is no guarantee that the derivative-free optimization methods will actually converge to $\mathbf{x}^*$.

The noise tolerance of these methods can be improved by accounting for the magnitude of the noise when comparing objective function values to accept or reject candidate vectors of design variables. The dynamic accuracy framework [6] uses bounds in the comparison of objective function values in order to specify the acceptable noise level that enables progress in the optimization. A challenge with this approach is that the conservative bounds may result in acceptable noise levels that decrease rapidly from one optimization iteration to the next, requiring increasingly large amounts of computational effort to compute the estimators $\hat{f}(\mathbf{x}_k)$, $\hat{g}(\mathbf{x}_k)$, and $\hat{h}(\mathbf{x}_k)$.

2.4 Numerical Results

We compare the efficiency of the multifidelity estimator against the regular Monte Carlo estimator with two example problems. The algorithm parameters are set to $n_{\text{init}} = 30$ and $n_\Delta = 10$. The examples in §2.4.1 and §2.4.2 are designed to illustrate the effect of the ratio of average computation time w and the correlation coefficient between the random output of the high-fidelity model and the random output of the low-fidelity model ρ_{AB}, respectively, on the computational cost of the multifidelity estimator.

Table 2.1: Distributions of the 5 random inputs for the short column example.

Random Variable	Distribution	Lower Bound	Upper Bound	Mean	Std. Dev.
$b(\omega)$	Uniform	5	15	–	–
$h(\omega)$	Uniform	15	25	–	–
$P(\omega)$	Normal	–	–	500	100
$M(\omega)$	Normal	–	–	2000	400
$Y(\omega)$	Lognormal	–	–	5	0.5

2.4.1 Short Column Uncertainty Propagation

This is a simple analytical example of a short column with rectangular cross-sectional area bh and yield stress Y subject to bending moment M and axial force P. The high-fidelity model is the limit-state function defined as [29]

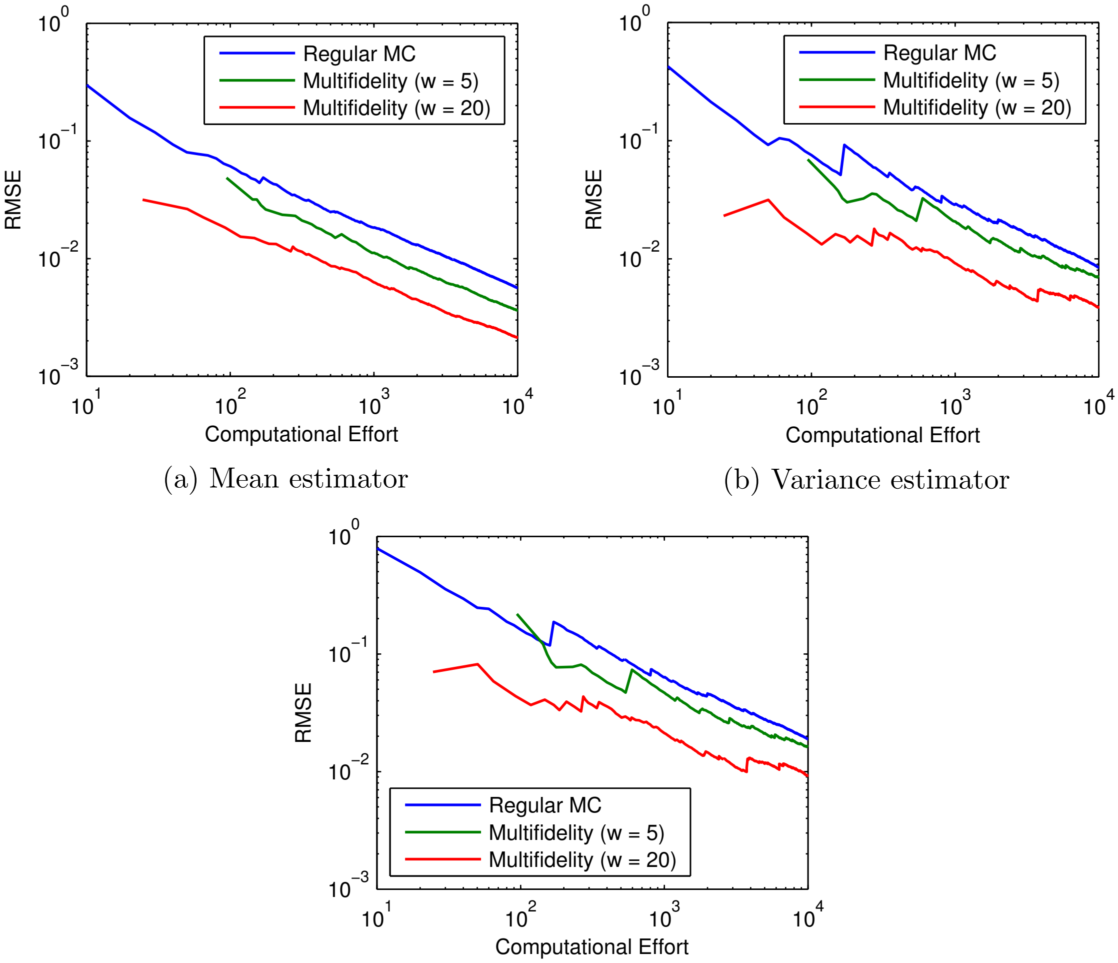
$$M_{\text{high}}(b,h,P,M,Y) = 1 - \frac{4M}{bh^2Y} - \left(\frac{P}{bhY}\right)^2$$

and a "low-fidelity model" is constructed by artificially perturbing the high-fidelity model:

$$M_{\text{low}}(b,h,P,M,Y) = 1 - \frac{3.8M}{bh^2Y} - \left(\frac{P\left(1+\frac{M-2000}{4000}\right)}{bhY}\right)^2.$$

For the purpose of demonstration, all five inputs are treated as random variables with distributions listed in Table 2.1. We seek to estimate $\mathbb{E}\left[M_{\text{high}}\right]$, $\operatorname{Var}\left[M_{\text{high}}\right]$ and $f = \mathbb{E}\left[M_{\text{high}}\right] + 3\sqrt{\operatorname{Var}\left[M_{\text{high}}\right]}$.

Both the high-fidelity model and the low-fidelity model take negligible time to evaluate, but we let the ratio of average computation times be $w = 5$ and $w = 20$ to see its effect on the variance reduction. The correlation coefficient between the high-fidelity model and the low-fidelity model (computed from (2.4c)) is $\hat{\rho}_{AB} = 0.984$ for the samples used to estimate the mean and is $\hat{\rho}_{AB} = 0.952$ for the samples used to estimate the variance. As discussed at the end of §2.1.2, when ρ_{AB} is close to 1, the amount of variance reduction is proportional to $1/w$ and hence we expect the RMSE in the multifidelity estimator to be approximately $1/\sqrt{w}$ that of the regular Monte Carlo estimator. We plot the RMSE (based on (2.3) and (2.5)) of the regular Monte Carlo estimators and the multifidelity estimators as a function of the computational



(a) Mean estimator

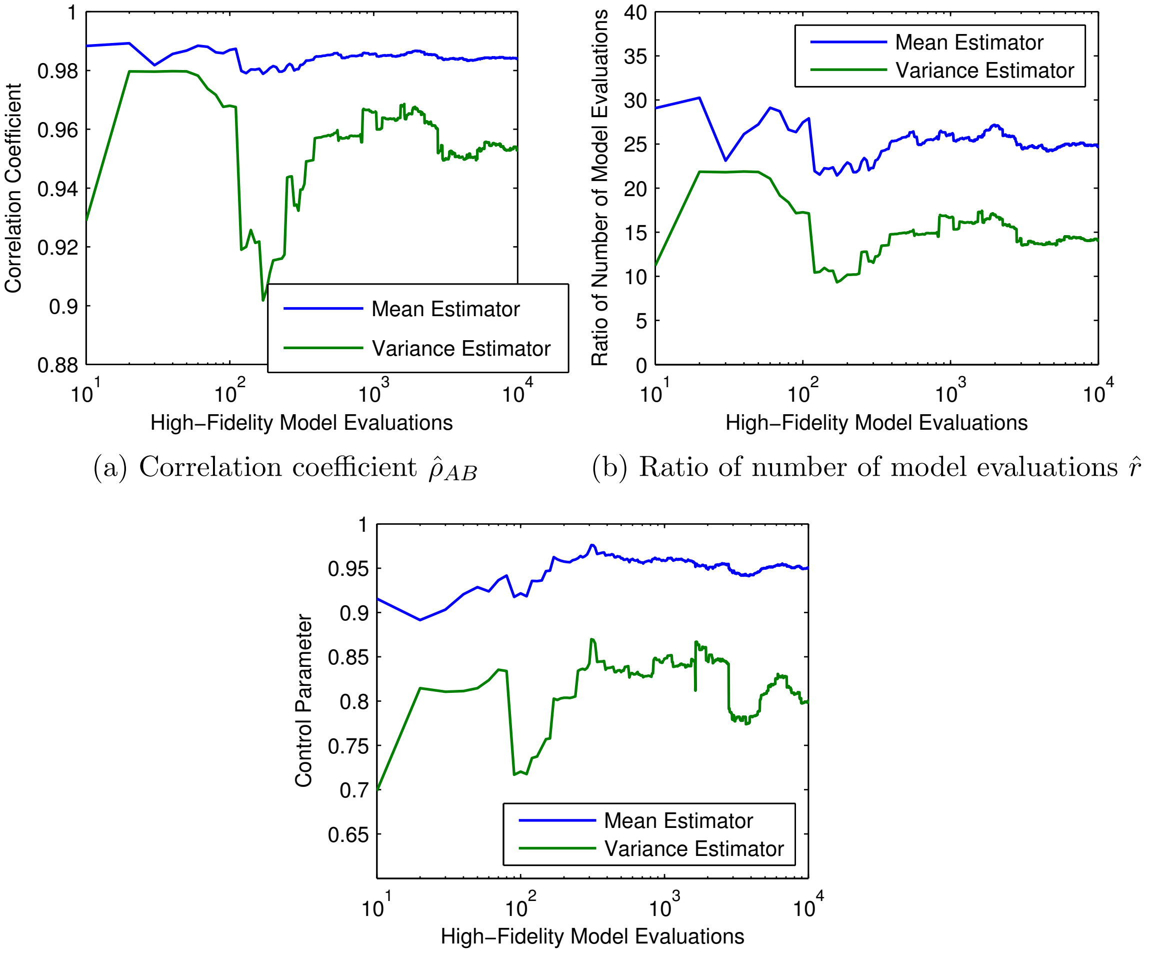
(b) Variance estimator

(c) $f = \mathbb{E}\left[M_{\text{high}}\right] + 3\sqrt{\text{Var}\left[M_{\text{high}}\right]}$ estimator

Figure 2-3: Root mean square error of the estimators as a function of the computational effort for the short column example.

effort p in Figure 2-3, where computational effort is the number of high-fidelity model evaluations for the regular Monte Carlo estimators and the equivalent number of high-fidelity model evaluations for the multifidelity estimators. The computational savings are significant. For example, to achieve 10^{-2} RMSE for the mean estimate, the regular Monte Carlo estimator requires about 3400 high-fidelity model evaluations, the multifidelity estimator with $w = 5$ requires about 1300 equivalent high-fidelity model evaluations, and the multifidelity estimator with $w = 20$ requires about 400 equivalent high-fidelity model evaluations.

In Figure 2-4, we plot the parameters $\hat{\rho}_{AB}$, $\hat{r}$ and $\hat{\alpha}$ used to compute the multifi-



(a) Correlation coefficient $\hat{\rho}_{AB}$

(b) Ratio of number of model evaluations $\hat{r}$

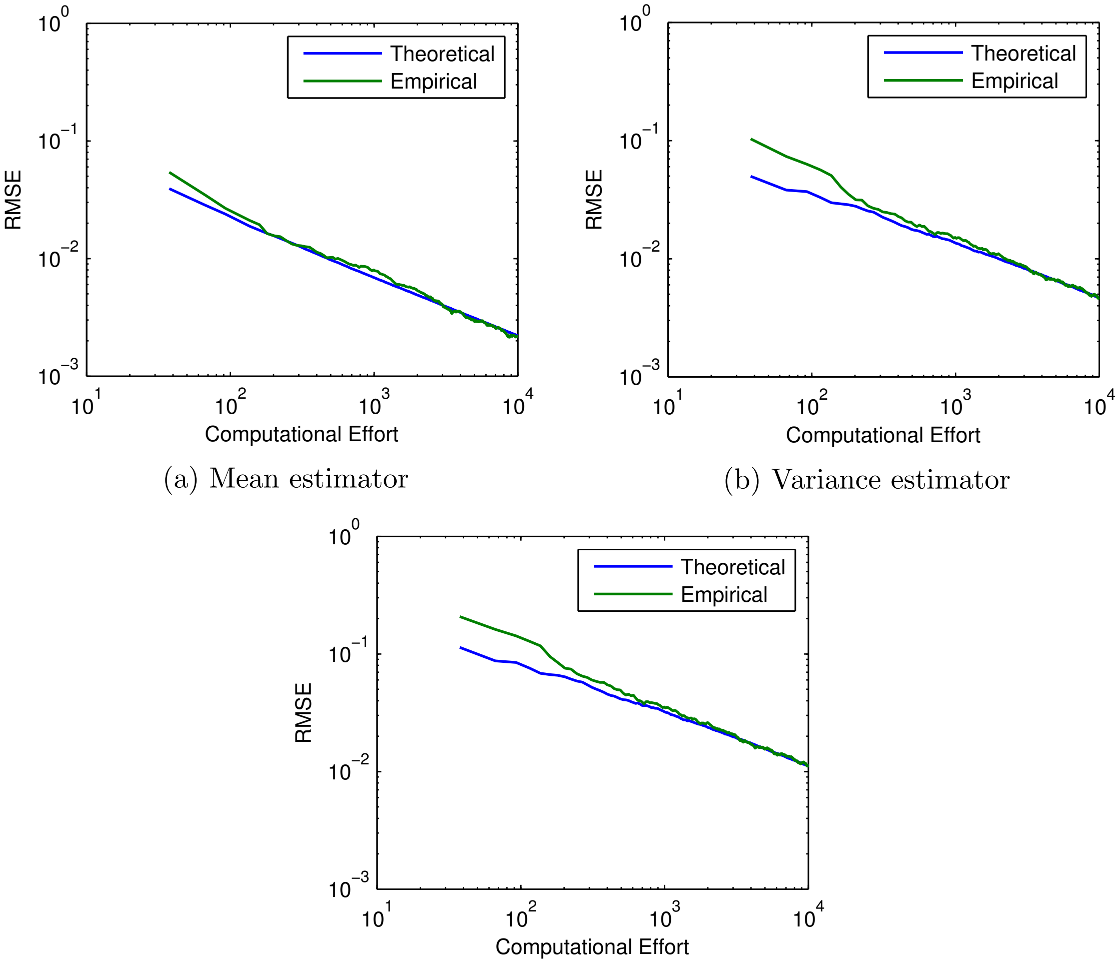
(c) Control parameter $\hat{\alpha}$

Figure 2-4: Parameters $\hat{\rho}_{AB}$, $\hat{r}$ and $\hat{\alpha}$ versus the number of high-fidelity model evaluations for the short column example with $w = 20$.

delity estimators. Their values fluctuate but, as discussed in §2.2.1, the multifidelity estimator is robust to errors in these parameters to a certain degree and we still obtain variance reduction as shown in Figure 2-3.

Finally, we verify our theoretical error for the multifidelity estimator from (2.3) and our approximation of the error for functions of statistics from (2.5). For $f = \mathbb{E}\left[M_{\text{high}}\right] + 3\sqrt{\text{Var}\left[M_{\text{high}}\right]}$, we have

$$\nabla f = \begin{bmatrix} 1 \\ \frac{3}{2\sqrt{\text{Var}\left[M_{\text{high}}\right]}} \end{bmatrix}.$$



(a) Mean estimator

(b) Variance estimator

(c) $f = \mathbb{E}\left[M_{\text{high}}\right] + 3\sqrt{\text{Var}\left[M_{\text{high}}\right]}$ estimator

Figure 2-5: Comparison of the theoretical root mean square errors to the empirical root mean square errors of the multifidelity estimators for the short column example with $w = 20$.

We compare the theoretical RMSE with empirical RMSE obtained by repeating the calculation with new realizations of the random inputs 100 times. The results are shown in Figure 2-5 and show good agreement. In particular, we observe that for functions of statistics that are not too nonlinear (as in this example), the RMSE approximation using (2.5) works well.

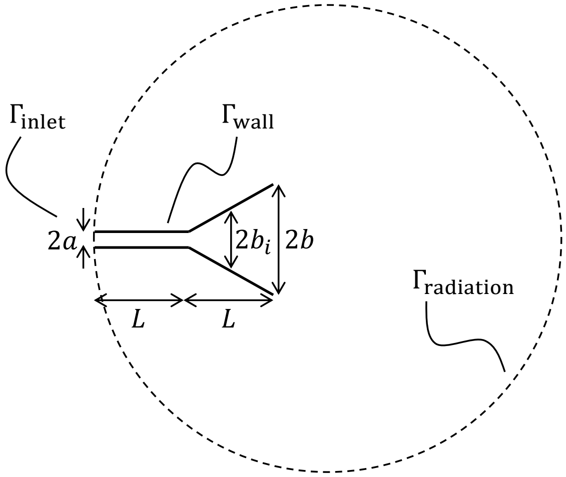


Figure 2-6: 2-D horn geometry where $a = 0.5$, $b = 3$, and $L = 5$. The shape of the horn flare is described by the half-widths b_i, $i = 1, \dots, 6$ uniformly distributed along the flare.

Table 2.2: Distributions of the 3 random inputs for the horn example.

Random Variable	Distribution	Lower Bound	Upper Bound	Mean	Std. Dev.
$k(\omega)$	Uniform	1.3	1.5	–	–
$z_u(\omega)$	Normal	–	–	50	3
$z_l(\omega)$	Normal	–	–	50	3

2.4.2 Acoustic Horn Uncertainty Propagation

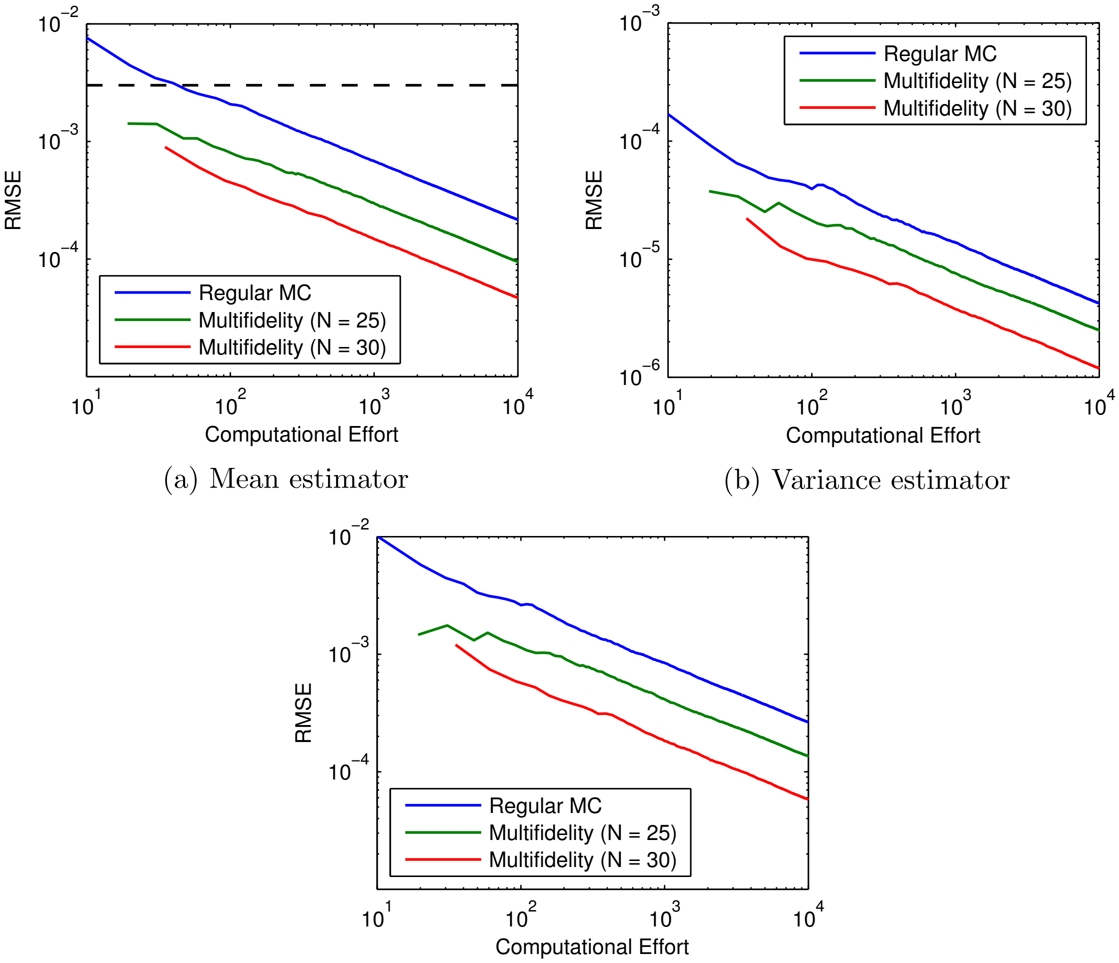
In this example, we model a 2-D acoustic horn governed by the non-dimensional complex Helmholtz equation $\nabla^2 u + k^2 u = 0$. An incoming wave enters the horn through the inlet and exits the outlet into the exterior domain with a truncated absorbing boundary $\Gamma_{\text{radiation}}$ [13]. The geometry of the horn is illustrated in Figure 2-6. The output of the model is the reflection coefficient $s = \left| \int_{\Gamma_{\text{inlet}}} u \, \mathrm{d}\Gamma - 1 \right|$, a measure of the horn's efficiency, and we again estimate its mean and variance. The three random inputs considered for this example are the wave number k, upper horn wall impedance z_u, and lower horn wall impedance z_l with distributions listed in Table 2.2. In the next section, we also consider six geometric parameters b_1 to b_6 describing the profile of the horn flare for optimization. Here, for uncertainty propagation, the geometric parameters are fixed at a straight flare profile as shown in Figure 2-6.

The high-fidelity model is a finite element model of the Helmholtz equation with

35,895 states and the low-fidelity model is a reduced basis model (with 9 parameters) constructed from the finite element discretization [43]. We consider two cases for the low-fidelity model: (i) a less accurate reduced basis model with $N = 25$ basis functions, and (ii) a more accurate reduced basis model with $N = 30$ basis functions. For the first case, the correlation coefficient between the high-fidelity model and the low-fidelity model (computed from (2.4c)) is $\hat{\rho}_{AB} = 0.959$ for the samples used to estimate the mean and is $\hat{\rho}_{AB} = 0.897$ for the samples used to estimate the variance. For the second case, it is $\hat{\rho}_{AB} = 0.998$ for the samples used to estimate the mean and is $\hat{\rho}_{AB} = 0.994$ for the samples used to estimate the variance. The increase in computational cost of the reduced basis model due to the five additional bases in the second case is negligible compared to the cost of the original finite element model and so the ratio of average computation times is $w = 40$ for both cases. This allows us to examine the effect of the correlation coefficient on the efficiency of the multifidelity estimator.

The RMSE of the mean estimator and of the variance estimator are shown in Figures 2-7a and 2-7b. We also plot the RMSE of $f = \mathbb{E}[s] + \sqrt{\mathrm{Var}[s]}$ in Figure 2-7c. The computational effort is the number of high-fidelity model evaluations for the regular Monte Carlo estimator and the equivalent number of high-fidelity model evaluations for the multifidelity estimator. This example demonstrates the benefit of a good correlation between the high-fidelity model and the low-fidelity model. To achieve 10^{-5} RMSE for the variance estimate, the regular Monte Carlo estimator requires about 1800 high-fidelity model evaluations, the multifidelity estimator with the less correlated low-fidelity model ($N = 25$ basis functions) requires about 600 equivalent high-fidelity model evaluations, and the multifidelity estimator with the more correlated low-fidelity model ($N = 30$ basis functions) requires about 100 equivalent high-fidelity model evaluations.

Given the high correlation between the reduced basis model with $N = 30$ basis functions and the finite element model, it appears to be simpler to throw all of the computational budget into computing a regular Monte Carlo estimator using only the reduced order model. If we do this for the mean estimate, there will be a bias



(a) Mean estimator

(b) Variance estimator

(c) $f = \mathbb{E}[s] + \sqrt{\mathrm{Var}[s]}$ estimator

Figure 2-7: Root mean square error of the estimators as a function of the computational effort for the horn example. The dashed line indicates the bias of the low-fidelity model ($N = 30$ basis functions).

of about -3×10^{-3} with respect to the true mean of the finite element model that cannot be reduced regardless of the number of cheap reduced order model evaluations used. For this problem, the bias is large relative to the RMSE that can be achieved by the multifidelity estimator (see Figure 2-7a) and highlights the fact that the choice of the low-fidelity model should be based on its correlation with the high-fidelity model rather than point-wise differences in the outputs. Furthermore, this suggests that the construction of the reduced basis model should be based on its correlation with respect to the finite element model.

We investigate the correlation between a reduced basis model and the finite element model for a simplified case. Let $M_{\text{high}}(\mathbf{u})$ be the finite element model and $M_{\text{low}}(\mathbf{u})$ be the reduced basis model, both with input parameters $\mathbf{u}$. Consider the L_2 error between the the two models weighted by the function $\pi(\mathbf{u})$:

$$\begin{aligned}\int_{-\infty}^{\infty} (M_{\text{high}}(\mathbf{u}) - M_{\text{low}}(\mathbf{u}))^2 \pi(\mathbf{u})\, \mathrm{d}\mathbf{u} = &-2\int_{-\infty}^{\infty} M_{\text{high}}(\mathbf{u}) M_{\text{low}}(\mathbf{u}) \pi(\mathbf{u})\, \mathrm{d}\mathbf{u} \\ &+ \int_{-\infty}^{\infty} M_{\text{low}}^2(\mathbf{u}) \pi(\mathbf{u})\, \mathrm{d}\mathbf{u} \\ &+ \int_{-\infty}^{\infty} M_{\text{high}}^2(\mathbf{u}) \pi(\mathbf{u})\, \mathrm{d}\mathbf{u}.\end{aligned}$$

Let the parameters $\mathbf{u}$ be a realization of the random variable $\mathbf{U}(\omega)$ with density function $\pi(\mathbf{u})$ and assume, for simplicity, that both the finite element model and the reduced basis model has zero mean. The weighted L_2 error then becomes

$$\begin{aligned}\mathbb{E}\left[(M_{\text{high}}(\mathbf{u}) - M_{\text{low}}(\mathbf{u}))^2\right] = &-2\,\mathrm{Cov}\left[M_{\text{high}}\mathbf{U}(\omega), M_{\text{low}}(\mathbf{U}(\omega))\right] \\ &+ \mathrm{Var}\left[M_{\text{low}}(\mathbf{U}(\omega))\right] + \mathrm{Var}\left[M_{\text{high}}(\mathbf{U}(\omega))\right].\end{aligned}$$

Thus, a reduced basis model construction strategy that (heuristically) minimizes its weighted L_2 error would attempt to reduce its variance and increase its covariance with respect to the finite element model, which can have the effect of increasing the correlation between the reduced basis model and the finite element model.

Table 2.3: Initial values, lower bounds, upper bounds, and optimal values of the 6 horn flare half-widths.

$\mathbf{b}_0$	$\mathbf{b}_L$	$\mathbf{b}_U$	$\mathbf{b}^*$
0.857	0.679	1.04	0.679
1.21	1.04	1.39	1.07
1.57	1.39	1.75	1.75
1.93	1.75	2.11	1.99
2.29	2.11	2.46	2.25
2.64	2.46	2.82	2.46

2.4.3 Acoustic Horn Robust Optimization

To demonstrate the effectiveness of the multifidelity estimator in the context of optimization under uncertainty, we consider the robust optimization of the shape of the acoustic horn flare. The design variables are $\mathbf{b} = [b_1 \cdots b_6]^\top$ representing the half-widths of the horn flare as shown in Figure 2-6. The initial values of the design variables (corresponding to the straight flare in §2.4.2), their lower bounds, and their upper bounds are listed in Table 2.3. The minimization of the horn reflection coefficient is formulated as

$$\min_{\mathbf{b}_L \le \mathbf{b} \le \mathbf{b}_U} f(\mathbf{b}) = \mathbb{E}\left[s(\mathbf{b}, \omega)\right] + \sqrt{\mathrm{Var}\left[s(\mathbf{b}, \omega)\right]}.$$

We employ the implicit filtering algorithm `imfil_v1` [23] developed for bound-constrained optimization problems without analytical derivatives. The algorithm calculates least-squares derivatives by evaluating the objective function in a stencil in the design space. A quasi-Newton search direction is generated from the least-squares derivatives and candidate vectors of design variables are accepted or rejected based on a line search. The optimization is conducted with the objective function evaluated using the regular Monte Carlo estimator and with the objective function evaluated using the multifidelity estimator. For the multifidelity estimator, the low-fidelity model is the reduced basis model with $N = 30$ basis functions. In both cases, the tolerance on the RMSE of the $f(\mathbf{b})$ estimator is fixed at 2×10^{-3}. Note that it also possible to specify a decreasing tolerance that depends on the size of the `imfil_v1` coordinate search stencil, but for simplicity it is not shown here. Three

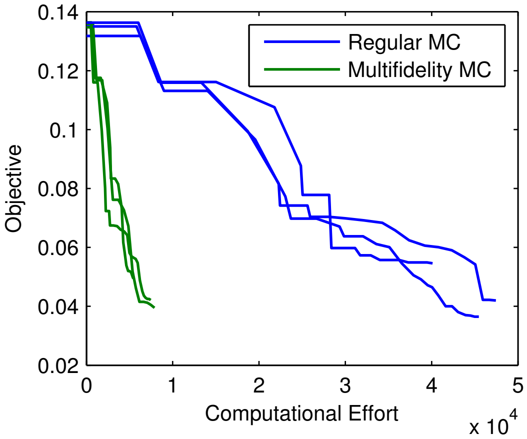
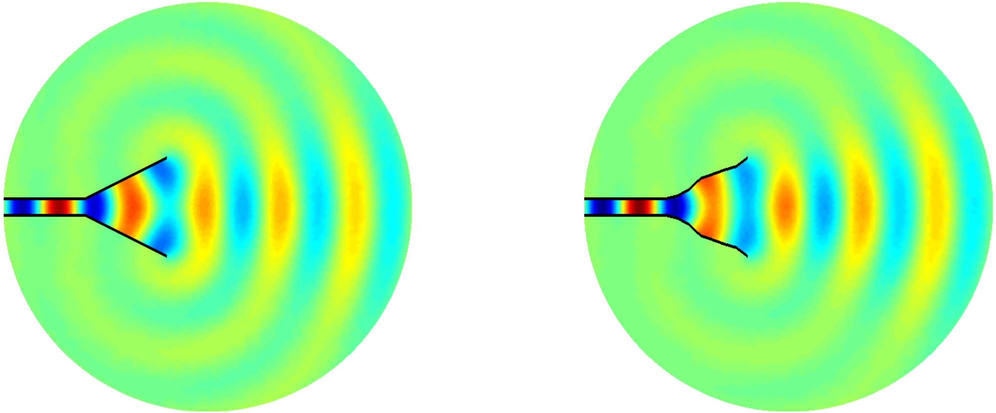


Figure 2-8: Comparison of convergence histories for the robust horn optimization using the regular Monte Carlo estimator and the multifidelity estimator. The optimization algorithm is implicit filtering.

trials were run for each case and the convergence of the objective as a function of the cumulative computational effort is shown in Figure 2-8. The computational effort is the number of high-fidelity model evaluations for the regular Monte Carlo estimator and the equivalent number of high-fidelity model evaluations for the multifidelity estimator. The computational savings are significant. It can be seen that using the multifidelity estimator provides cumulative computational savings over the course of the optimization and we locate the optimum using about 85% less computational effort. The solution field for the initial straight horn and the optimal horn are shown in Figure 2-9. The mean reflection coefficient at the initial shape is 0.113 with a standard deviation of 0.0228 and the mean reflection coefficient at the optimal shape is 0.0220 with a standard deviation of 0.0124.

2.5 Chapter Summary

In this chapter, we presented a multifidelity approach to estimate statistics of the output of an expensive high-fidelity model. We showed that employing an inexpensive low-fidelity model whose output is correlated with that of the high-fidelity model can reduce the computational cost, measured in terms of the equivalent number of high-fidelity model evaluations, to achieve a desired error tolerance in the statistic estimates



(a) Initial shape

(b) Optimal shape

Figure 2-9: Finite element solution of the Helmholtz equation at the initial horn design $\mathbf{b}_0$ and robust optimal horn design $\mathbf{b}^*$.

relative to regular Monte Carlo simulation. Numerical results for the acoustic horn robust optimization example demonstrated 85% reduction in computational cost.

In practice, we do not always have the luxury of a good low-fidelity model to compute the multifidelity estimator. In the next chapter, we consider how to perform optimization under uncertainty using only the high-fidelity model without incurring excessive computational cost.

Chapter 3

Leveraging Model Correlation Over Design Space

In this chapter, we develop an alternative method to compute the estimator $\hat{s}_A(\mathbf{x})$ of the exact statistic $s_A(\mathbf{x})$. In §3.1, we consider the situation where we do not have an inexpensive low-fidelity model and introduce the information reuse estimator that makes use of another source of approximate information available during optimization under uncertainty. Next, we discuss implementation issues in §3.2. In §3.3, we return to the situation where a low-fidelity model is available and combine the multifidelity estimator and the information reuse estimator to further improve efficiency. Finally, we demonstrate the effectiveness of the information reuse estimator and the combined estimator with numerical examples in §3.4.

3.1 Approach

Consider the problem of estimating statistics to evaluate objective (and constraint) functions for optimization, such as problem (1.1). We cannot compute the multifidelity estimator described in Chapter 2 because we do not have a low-fidelity model. Fortunately, during optimization under uncertainty, the statistics of interest are computed at many different vectors of design variables, generating an alternative source of approximate information. This is useful information because the random output of the model at one vector of design variables is often correlated with the random output of the model at another vector of design variables.

3.1.1 Model Autocorrelation over Design Space

The model output $M(\mathbf{x}, \mathbf{U}(\omega))$ can be interpreted as a random process indexed by the vector of design variables $\mathbf{x}$. Therefore, if $A(\omega)$ is the random output of the model at a particular vector of design variables and $C(\omega)$ is the random output of the model at another vector of design variables, then the autocorrelation structure of $M(\mathbf{x}, \mathbf{U}(\omega))$ provides the correlation between $A(\omega)$ and $C(\omega)$ needed to make $C(\omega)$ a suitable auxiliary random variable for the control variate method. Intuitively, if the model $M(\mathbf{x}, \mathbf{u})$ at a realization $\mathbf{u}$ of the random input vector $\mathbf{U}(\omega)$ is smooth in the $\mathbf{x}$ direction, then a small perturbation $\mathbf{x} + \Delta\mathbf{x}$ produces only a small change in the output. Considering all realizations of $\mathbf{U}(\omega)$, it is reasonable to think that $M(\mathbf{x} + \Delta\mathbf{x}, \mathbf{U}(\omega))$ is correlated with $M(\mathbf{x}, \mathbf{U}(\omega))$

To make the argument concrete, we determine an approximation of the autocorrelation of $M(\mathbf{x}, \mathbf{U}(\omega))$ for the simpler case of a scalar design variable x. Let the model $M(x, \mathbf{u})$ be twice differentiable in x for all realizations $\mathbf{u}$ of $\mathbf{U}(\omega)$. Applying a second order Taylor expansion in x, the correlation coefficient between $M(x + \Delta x, \mathbf{U}(\omega))$ and $M(x, \mathbf{U}(\omega))$ is quadratic in Δx for $|\Delta x| \ll 1$ (see derivation in Appendix A):

$$\begin{aligned} &\operatorname{Corr}\left[M(x+\Delta x, \mathbf{U}(\omega)), M(x, \mathbf{U}(\omega)))\right] \\ &\approx 1 - \frac{1 - \operatorname{Corr}\left[M'(x, \mathbf{U}(\omega)), M(x, \mathbf{U}(\omega))\right]^2}{2 \operatorname{Var}\left[M(x, \mathbf{U}(\omega))\right] / \operatorname{Var}\left[M'(x, \mathbf{U}(\omega))\right]} \Delta x^2, \end{aligned} \tag{3.1}$$

where $M'(x, \mathbf{u}) = \frac{\partial M(x, \mathbf{u})}{\partial x}$. We see that $\operatorname{Corr}\left[M(x+\Delta x, \mathbf{U}(\omega)), M(x, \mathbf{U}(\omega)))\right] \to 1$ as $\Delta x \to 0$. The numerator of the expression shows that the correlation also improves when the change in model output is (positively or negatively) proportional to the model output across all realizations of $\mathbf{U}(\omega)$. To illustrate why this is the case, consider a large number of samples $M(x, \mathbf{u}_i)$, $i = 1, 2, 3, \ldots$ sorted in ascending order. A change that is proportional to $M(x, \mathbf{u}_i)$ allows the samples $M(x + \Delta x, \mathbf{u}_i)$, $i =$

$1, 2, 3, \ldots$ to remain in ascending order. If the samples are no longer in ascending order, then clearly the correlation is degraded.

If we let $A(\omega)$ be the random output of the model at $\mathbf{x}$ and let $C(\omega)$ be the random output of the model at $\mathbf{x}+\Delta\mathbf{x}$, then, based on (3.1), we can maximize the correlation between $A(\omega)$ and $C(\omega)$ by choosing $\|\Delta\mathbf{x}\|$ to be as small as possible. This motivates the use of the information reuse estimator within an optimization algorithm, where a sequence of potentially small steps in the design variables readily provides good candidates for the choice of the auxiliary random variable $C(\omega)$.

3.1.2 Information Reuse Estimator

Let k be the current optimization iteration and let $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$ be the sequence of design variables visited by the optimization algorithm. We define the random variable $A(\omega) = M(\mathbf{x}_k, \mathbf{U}(\omega))$ and we wish to compute an estimator $\hat{s}_{A,p}$ of the exact statistic $s_A = \mathbb{E}\left[A(\omega)\right]$ and the estimator variance $\mathrm{Var}\left[\hat{s}_{A,p}\right]$, where the computational effort p is to be defined below. Furthermore, we define the auxiliary random variable $C(\omega) = M(\mathbf{x}_\ell, \mathbf{U}(\omega))$ for $\ell < k$. We assume that during optimization iteration ℓ, we have stored the then current estimator and its estimator variance in a database. Therefore, at the current optimization iteration k, we have available the estimator $\hat{s}_C$ of the exact statistic $s_C = \mathbb{E}\left[C(\omega)\right]$ as well as the estimator variance $\mathrm{Var}\left[\hat{s}_C\right]$.[1] Analogous to the multifidelity estimator, we do not know the exact statistic s_C of the auxiliary random variable $C(\omega)$ required by the control variate method. Therefore, we replace it with $\hat{s}_C$ and obtain information reuse estimator of s_A, denoted as $\hat{s}_{A,p}$, as

$$\hat{s}_{A,p} = \bar{a}_n + \gamma(\hat{s}_C - \bar{c}_n). \tag{3.2}$$

[1] We do not indicate the computation effort of the estimator $\hat{s}_C$ because it is not relevant to the current optimization iteration.

for some control parameter $\gamma \in \mathbb{R}$. The variance of the information reuse estimator is

$$\begin{aligned}\mathrm{Var}\left[\hat{s}_{A,p}\right] &= \mathrm{Var}\left[\bar{a}_n\right] + \gamma^2(\mathrm{Var}\left[\hat{s}_C\right] + \mathrm{Var}\left[\bar{c}_n\right]) - 2\gamma\,\mathrm{Cov}\left[\bar{a}_n, \bar{c}_n\right] \\ &= \frac{\sigma_A^2}{n} + \gamma^2\left(\mathrm{Var}\left[\hat{s}_C\right] + \frac{\sigma_C^2}{n}\right) - 2\gamma\frac{\rho_{AC}\sigma_A\sigma_C}{n} \\ &= \frac{1}{n}\left[\sigma_A^2 + \gamma^2\sigma_C^2(1+\eta) - 2\gamma\rho_{AC}\sigma_A\sigma_C\right].\end{aligned}$$

where $\eta = n\,\mathrm{Var}\left[\hat{s}_C\right]/\sigma_C^2$. Note that this formulation implies that $\hat{s}_C$ is uncorrelated with $\bar{a}_n$ or $\bar{c}_n$. This can be achieved in practice by ensuring that the set of realizations of $\mathbf{U}(\omega)$ used in optimization iteration k is independent of the set of realizations of $\mathbf{U}(\omega)$ used in optimization iteration ℓ. Otherwise, additional covariance terms, $\mathrm{Cov}\left[\bar{a}_n, \hat{s}_C\right]$ and $\mathrm{Cov}\left[\hat{s}_C, \bar{c}_n\right]$, appear in the expression. This is problematic because $\hat{s}_C$ is itself the information reuse estimator at optimization iteration ℓ with its own auxiliary random variable at yet another previous optimization iteration and so on, resulting in a chain of covariance terms between random variables stretching back to the first optimization iteration. By requiring that $\hat{s}_C$ is independent of $\bar{a}_n$ or $\bar{c}_n$, we break this chain of dependence and simplify the expression.

Computing the information reuse estimator at optimization iteration k using the samples $\mathbf{u}_i$, $i = 1, 2, 3, \ldots, n$ drawn from the distribution of the random input vector $\mathbf{U}(\omega)$ requires n model evaluations at $\mathbf{x}_k$ to calculate $\bar{a}_n$ and n model evaluations at $\mathbf{x}_\ell$ to calculate $\bar{c}_n$. Therefore, the computational effort is $p = 2n$. Given a fixed computational budget p, we minimize $\mathrm{Var}\left[\hat{s}_{A,p}\right]$ in terms of γ and obtain

$$\gamma^* = \left(\frac{\rho_{AC}}{1+\eta}\right)\frac{\sigma_A}{\sigma_C}$$

and

$$\mathrm{MSE}\left[\hat{s}_{A,p}^*\right] = \mathrm{Var}\left[\hat{s}_{A,p}^*\right] = 2\left(1 - \frac{\rho_{AC}^2}{1+\eta}\right)\frac{\sigma_A^2}{p}. \tag{3.3}$$

Therefore, the information reuse estimator has a low variance when the correlation between the model output at $\mathbf{x}_k$ and the model output at $\mathbf{x}_\ell$ is high. As discussed in

§3.1.1, this is likely to happen if we select $\ell = \arg\min_{\ell' < k} \|\mathbf{x}_k - \mathbf{x}_{\ell'}\|$. However, the amount of variance reduction is degraded when the parameter η is large, which occurs when $\mathrm{Var}\left[\hat{s}_C\right]$ is large relative to $\mathrm{Var}\left[\bar{c}_n\right] = \sigma_C^2/n$. This indicates that the information reuse estimator may not be beneficial in terms of computational cost if the desired estimator variance $\mathrm{Var}\left[\hat{s}^*_{A,p}\right]$ at the current optimization iteration is much lower than the estimator variance at the previous optimization iteration, $\mathrm{Var}\left[\hat{s}_C\right]$.

Unlike the variance of the original control variate estimator or the variance of the multifidelity estimator, the variance of the information reuse estimator can potentially be higher than that of the regular Monte Carlo estimator. When this occurs, we switch to the regular Monte Carlo estimator as a safeguard. Fortunately, the results in §3.4 suggest that the correlation ρ_{AC} is often high enough to prevent this occurrence.

3.1.3 Correlated Estimator Errors

Since the results of an optimization iteration are reused at a later optimization iteration, the information reuse estimators computed during the course of optimization are correlated with each other. To illustrate, we derive an expression for the correlation coefficient between $\hat{s}_{A,p}$, the estimator at optimization iteration k, and $\hat{s}_C$, the estimator at optimization iteration $\ell < k$. Since $\mathrm{Cov}\left[\bar{a}_n, \hat{s}_C\right] = \mathrm{Cov}\left[\bar{c}_n, \hat{s}_C\right] = 0$, we have

$$\begin{aligned}
\mathrm{Cov}\left[\hat{s}_{A,p}, \hat{s}_C\right] &= \mathrm{Cov}\left[\bar{a}_n + \gamma(\hat{s}_C - \bar{c}_n), \hat{s}_C\right] \\
&= \gamma\, \mathrm{Cov}\left[\hat{s}_C, \hat{s}_C\right] \\
&= \left(\frac{\rho_{AC}}{1+\eta}\right)\frac{\sigma_A}{\sigma_C}\, \mathrm{Var}\left[\hat{s}_C\right].
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\mathrm{Var}\left[\hat{s}_{A,p}\right] \mathrm{Var}\left[\hat{s}_C\right] &= \left(1 - \frac{\rho_{AC}^2}{1+\eta}\right)\frac{\sigma_A^2}{n}\, \mathrm{Var}\left[\hat{s}_C\right] \\
&= \left(\frac{1+\eta-\rho_{AC}^2}{1+\eta}\right)\frac{\sigma_A^2}{n}\, \mathrm{Var}\left[\hat{s}_C\right].
\end{aligned}$$

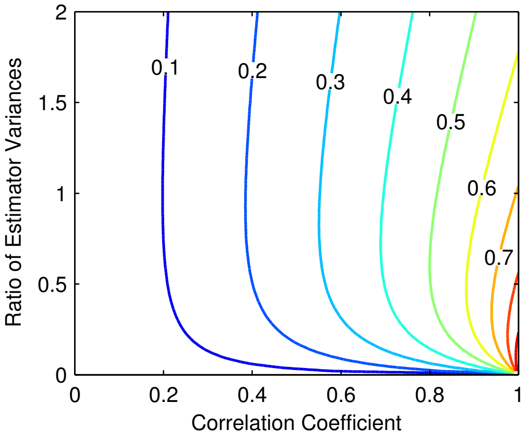


Figure 3-1: A contour plot of the correlation of the estimators $\operatorname{Corr}[\hat{s}_{A,p}, \hat{s}_C]$ as a function of the correlation coefficient of the model outputs ρ_{AC} and the ratio of estimator variances $\eta = \operatorname{Var}[\hat{s}_C]/\operatorname{Var}[\bar{c}_n]$.

Therefore,

$$\begin{aligned}
\operatorname{Corr}[\hat{s}_{A,p}, \hat{s}_C] &= \frac{\operatorname{Cov}[\hat{s}_{A,p}, \hat{s}_C]}{\sqrt{\operatorname{Var}[\hat{s}_{A,p}]\operatorname{Var}[\hat{s}_C]}} \\
&= \rho_{AC}\sqrt{\frac{n\operatorname{Var}[\hat{s}_C]}{(1+\eta)(1+\eta-\rho_{AC}^2)\sigma_C^2}} \\
&= \rho_{AC}\sqrt{\frac{\eta}{(1+\eta)(1+\eta-\rho_{AC}^2)}} \\
&= \frac{\rho_{AC}}{\sqrt{(1+1/\eta)(1+\eta-\rho_{AC}^2)}}. \qquad (3.4)
\end{aligned}$$

A contour plot of the correlation of the estimators $\operatorname{Corr}[\hat{s}_{A,p}, \hat{s}_C]$ as a function of the correlation coefficient of the model outputs ρ_{AC} and the ratio of estimator variances $\eta = \operatorname{Var}[\hat{s}_C]/\operatorname{Var}[\bar{c}_n]$ is shown in Figure 3-1. As the model outputs become more correlated, i.e., ρ_{AC} is high, the estimators become more correlated as well.

If we assume normality, then $\hat{s}_{A,p}$ and $\hat{s}_C$ are jointly normally distributed as

$$\begin{bmatrix} \hat{s}_{A,p} \\ \hat{s}_C \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} s_A \\ s_C \end{bmatrix}, \begin{bmatrix} \operatorname{Var}[\hat{s}_{A,p}] & \operatorname{Cov}[\hat{s}_{A,p}, \hat{s}_C] \\ \operatorname{Cov}[\hat{s}_{A,p}, \hat{s}_C] & \operatorname{Var}[\hat{s}_C] \end{bmatrix}\right),$$

where $\operatorname{Var}[\hat{s}_{A,p}]$ is given by (3.3). Thus, although $\hat{s}_{A,p}$ is an unbiased estimator of

s_A,[2] $\hat{s}_{A,p}$ conditioned on $\hat{s}_C$ is biased:

$$\hat{s}_{A,p} \mid \hat{s}_C \sim \mathcal{N}\Bigg(s_A + \mathrm{Corr}\left[\hat{s}_{A,p}, \hat{s}_C\right] \sqrt{\frac{\mathrm{Var}\left[\hat{s}_{A,p}\right]}{\mathrm{Var}\left[\hat{s}_C\right]}} (\hat{s}_C - s_C), \\ \left(1 - \mathrm{Corr}\left[\hat{s}_{A,p}, \hat{s}_C\right]^2\right) \mathrm{Var}\left[\hat{s}_{A,p}\right] \Bigg).$$

In other words, given a realization of the estimator at the first optimization iteration, the information reuse estimators at subsequent optimization iterations are biased. For fixed values of $\mathrm{Var}\left[\hat{s}_{A,p}\right]$ and $\mathrm{Var}\left[\hat{s}_C\right]$, as $\mathrm{Corr}\left[\hat{s}_{A,p}, \hat{s}_C\right]$ increases, the estimator $\hat{s}_{A,p}$ conditioned on a given value of $\hat{s}_C$ trades less estimator variance for more bias. Thus, the information reuse estimator can have a smoothing effect on the noise of the objective and constraint functions during optimization under uncertainty. This can impact the selection and behaviour of the optimization algorithm and is discussed further in §3.4.1.

3.2 Implementation

Many aspects of the implementation of the information reuse estimator are similar to those of the multifidelity estimator discussed in §2.2. The main difference is that the procedure to compute the information reuse estimator is necessarily embedded within an outer optimization loop. We also provide details on the safeguard mechanism that prevents the information reuse estimator from requiring more computational effort than the regular Monte Carlo estimator to meet a given error tolerance.

3.2.1 Parameter Calculations

In practice, since σ_A, σ_C, and ρ_{AC} are unknown, the optimal parameters γ^* and the parameter η are replaced by their sample estimates $\hat{\gamma}$ and $\hat{\eta}$ based on the n samples

[2] $\mathbb{E}\left[\hat{s}_{A,p}\right] = \mathbb{E}\left[\bar{a}_n\right] + \gamma(\mathbb{E}\left[\hat{s}_C\right] - \mathbb{E}\left[\bar{c}_n\right]) = s_A + \gamma(s_C - s_C) = s_A$.

of $A(\omega)$ and $C(\omega)$, $\{a_i, c_i\}_{i=1}^n$:

$$\hat{\gamma} = \frac{1}{1+\hat{\eta}} \frac{\sum_{i=1}^n (a_i - \bar{a}_n)(c_i - \bar{c}_n)}{\sum_{i=1}^n (c_i - \bar{c}_n)^2}, \tag{3.5a}$$

$$\hat{\eta} = \frac{\mathrm{Var}\,[\hat{s}_C] n(n-1)}{\sum_{i=1}^n (c_i - \bar{c}_n)^2}, \tag{3.5b}$$

$$\hat{\rho}_{AC}^2 = \frac{\left[\sum_{i=1}^n (a_i - \bar{a}_n)(c_i - \bar{c}_n)\right]^2}{\left[\sum_{i=1}^n (a_i - \bar{a}_n)^2\right]\left[\sum_{i=1}^n (c_i - \bar{c}_n)^2\right]}, \tag{3.5c}$$

$$\hat{\sigma}_A^2 = \frac{\sum_{i=1}^n (a_i - \bar{a}_n)^2}{n-1}. \tag{3.5d}$$

3.2.2 Safeguard Mechanism

There is a chance, especially during the first few optimization iterations when the optimizer tends to take larger steps and there are fewer candidates for the choice of ℓ, that the correlation between the random model output at $\mathbf{x}_k$ and the random model output at $\mathbf{x}_\ell$ is low. When this occurs, the information reuse estimator may require more computational effort than the regular Monte Carlo estimator to meet a desired error tolerance. The safeguard mechanism detects this situation and falls back to the regular Monte Carlo estimator.

From the initial set of n_{init} samples $\{a_i, c_i\}_{i=1}^{n_{\text{init}}}$, we calculate the parameters for the information reuse estimator from (3.5) and compute the initial information reuse estimator $\hat{s}_{A,p_{\text{init}}}$ from (3.2), where $p_{\text{init}} = 2n_{\text{init}}$. Using the expressions for the mean square error (3.3) and (2.1), we can calculate the computational effort needed by the information reuse estimator and the regular Monte Carlo estimator, respectively, to meet the desired RMSE. If the information reuse estimator requires less computational effort than the regular Monte Carlo estimator, we increment the number of samples as needed and continue to compute the information reuse estimator without change. Otherwise, we switch and compute the regular Monte Carlo estimator using the samples a_i, $i = n_{\text{init}}+1, n_{\text{init}}+2, n_{\text{init}}+3, \ldots$. In other words, we stop evaluating the model at $\mathbf{x}_\ell$.

In the safeguarded case, we obtain the regular Monte Carlo estimator $\bar{a}_{p_{\text{safe}}}$ using computational effort $p_{\text{safe}} = n - n_{\text{init}}$. However, there is no reason to discard the initial information reuse estimator $\hat{s}_{A,p_{\text{init}}}$ in which we have already expended computational effort $p_{\text{init}} = 2n_{\text{init}}$ to compute. Therefore, we take a linear combination of the two with parameter $\theta \in \mathbb{R}$, $0 \leq \theta \leq 1$ to produce the safeguarded information reuse estimator, denoted as $\hat{s}_{A,p}$ [36]:

$$\hat{s}_{A,p} = \theta \hat{s}_{A,p_{\text{init}}} + (1-\theta)\bar{a}_{p_{\text{safe}}} \tag{3.6}$$

with computational effort $p = p_{\text{init}} + p_{\text{safe}} = n_{\text{init}} + n$. The variance of the safeguarded information reuse estimator is

$$\operatorname{Var}\left[\hat{s}_{A,p}\right] = \theta^2 \operatorname{Var}\left[\hat{s}_{A,p_{\text{init}}}\right] + (1-\theta)^2 \operatorname{Var}\left[\bar{a}_{p_{\text{safe}}}\right],$$

where $\operatorname{Var}\left[\hat{s}_{A,p_{\text{init}}}\right]$ is evaluated from (3.3) and $\operatorname{Var}\left[\bar{a}_{p_{\text{safe}}}\right]$ is evaluated from (2.1). It is minimized when

$$\theta^* = \frac{\operatorname{Var}\left[\bar{a}_{p_{\text{safe}}}\right]}{\operatorname{Var}\left[\hat{s}_{A,p_{\text{init}}}\right] + \operatorname{Var}\left[\bar{a}_{p_{\text{safe}}}\right]},$$

resulting in

$$\operatorname{MSE}\left[\hat{s}^*_{A,p}\right] = \operatorname{Var}\left[\hat{s}^*_{A,p}\right] = \frac{\operatorname{Var}\left[\hat{s}_{A,p_{\text{init}}}\right] \operatorname{Var}\left[\bar{a}_{p_{\text{safe}}}\right]}{\operatorname{Var}\left[\hat{s}_{A,p_{\text{init}}}\right] + \operatorname{Var}\left[\bar{a}_{p_{\text{safe}}}\right]}. \tag{3.7}$$

The optimal safeguard parameter θ^* puts more weight on either the initial information reuse estimator $\hat{s}_{A,p_{\text{init}}}$ or the regular Monte Carlo estimator $\bar{a}_{p_{\text{safe}}}$ depending on which one has a lower estimator variance. The optimal linear combination allows the safeguarded information reuse estimator $\hat{s}_{A,p}$ to have a lower estimator variance than either than initial information reuse estimator $\hat{s}_{A,p_{\text{init}}}$ or the regular Monte Carlo estimator $\bar{a}_{p_{\text{safe}}}$.

3.2.3 Iterative Procedure

We first consider the procedure to compute the information reuse estimator for the exact statistic $s_A = \mathbb{E}\left[A(\omega)\right] = \mathbb{E}\left[M(\mathbf{x}_k, \mathbf{U}(\omega))\right]$ at a given vector of design variables

$\mathbf{x}_k$ in Algorithm 3.1. The estimator $\hat{s}_C$ and its variance $\mathrm{Var}\left[\hat{s}_C\right]$ are also given. The procedure begins with a set of $n = n_{\text{init}}$ samples $\{a_i, c_i\}_{i=1}^{n}$. This initial set of samples serves an important purpose—determining whether to (i) continue with the information reuse estimator or (ii) switch to the safeguarded information reuse estimator. Once this is determined, n is incremented by n_{Δ} every iteration and the procedure is similar to that the of multifidelity estimator. Omitting algorithm overhead, the computational expense is $p = 2n$ for the first case or $p = n_{\text{init}} + n$ for the second case.

Since Algorithm 3.1 requires a given vector of design variables $\mathbf{x}_\ell$ and the corresponding estimator $\hat{s}_C$ with variance $\mathrm{Var}\left[\hat{s}_C\right]$, it is generally used within an outer optimization loop[3] as shown in Algorithm 3.2. At the initial vector of design variables $\mathbf{x}_0$, we cannot compute the information reuse estimator and thus we start with the regular Monte Carlo estimator. We save the estimator and its variance computed during every optimization iteration in a database to provide candidates for $\hat{s}_C$ and $\mathrm{Var}\left[\hat{s}_C\right]$ at subsequent optimization iterations. As discussed in §3.1.2, at each optimization iteration, a new (independent from previous optimization iterations) stream of random input vectors $\mathbf{u}_i$, $i = 1, 2, 3, \ldots$ is used to evaluate the model and compute the estimators. As a result, the information reuse estimator is incompatible with the sample average approximation method described in §2.3.2 in the outer optimization loop.

[3] Technically, the information reuse estimator can be used within any outer loop that generates a sequence of vectors of design variables. However, it is most effective when the vectors of design variables cluster together, as is often the case with optimization when it is close to the optimum.

Algorithm 3.1 Information Reuse Estimator

```
Given desired RMSE, design variables $\mathbf{x}_k$ and $\mathbf{x}_\ell$, estimator $\hat{s}_C$ and its variance $\mathrm{Var}[\hat{s}_C]$, initial number of samples $n_{\text{init}}$, increment in number of samples $n_\Delta$, model $M(\mathbf{x}, \mathbf{u})$, and a sequence of pseudo-random input vectors $\mathbf{u}_i$ for $i = 1, 2, 3, \dots$ drawn from the distribution of $\mathbf{U}(\omega)$:
 1  Evaluate samples $a_i = M(\mathbf{x}_k, \mathbf{u}_i)$ and $c_i = M(\mathbf{x}_\ell, \mathbf{u}_i)$ for $i = 1, \dots, n_{\text{init}}$.
 2  Compute $\bar{a}_{n_{\text{init}}}$ using $\{a_i\}_{i=1}^{n_{\text{init}}}$ and $\bar{c}_{n_{\text{init}}}$ using $\{c_i\}_{i=1}^{n_{\text{init}}}$.
 3  Compute $\hat{\gamma}$, $\hat{\eta}$, $\hat{\rho}_{AC}^2$, and $\hat{\sigma}_A^2$ from (3.5) using $\{a_i, c_i\}_{i=1}^{n_{\text{init}}}$.
 4  Compute the initial information reuse estimator from (3.2).
 5  Estimate computational effort $p_1$ for information reuse estimator to meet desired RMSE from (3.3).
 6  Estimate computational effort $p_2$ for the safeguarded information reuse estimator to meet desired RMSE from (3.7).
 7  Let $n_{\text{old}} = n_{\text{init}}$ and $n = n_{\text{init}} + n_\Delta$.
 8  If $p_1 > p_2$, skip ahead to Step 16; otherwise, continue to Step 9.
 9  Continuing on with the information reuse estimator:
10  Evaluate samples $a_i = M(\mathbf{x}_k, \mathbf{u}_i)$ and $c_i = M(\mathbf{x}_\ell, \mathbf{u}_i)$ for $i = n_{\text{old}} + 1, \dots, n$.
11  Compute $\bar{a}_n$ using $\{a_i\}_{i=1}^{n}$ and $\bar{c}_n$ using $\{c_i\}_{i=1}^{n}$.
12  Compute $\hat{\gamma}$, $\hat{\eta}$, $\hat{\rho}_{AC}^2$, and $\hat{\sigma}_A^2$ from (3.5) using $\{a_i, c_i\}_{i=1}^{n}$.
13  Compute information reuse estimator from (3.2).
14  Compute RMSE from (3.3).
15  If RMSE is too large, set $n_{\text{old}} = n$, $n \leftarrow n + n_\Delta$ and return to Step 9; otherwise, stop.
16  Continuing on with the safeguarded information reuse estimator:
17  Evaluate samples $a_i = M(\mathbf{x}_k, \mathbf{u}_i)$ for $i = n_{\text{old}} + 1, \dots, n$.
18  Compute $\bar{a}_{p_{\text{safe}}}$ using $\{a_i\}_{i=n_{\text{init}}+1}^{n}$.
19  Compute $\hat{\sigma}_A^2$ from (3.5d) using $\{a_i\}_{i=n_{\text{init}}+1}^{n}$.
20  Compute the safeguarded information reuse estimator from (3.6).
21  Compute RMSE from (3.7).
22  If RMSE is too large, set $n_{\text{old}} = n$, $n \leftarrow n + n_\Delta$ and return to Step 16; otherwise, stop.
```

Algorithm 3.2 Optimization Using Information Reuse Estimator

```
Given an optimization algorithm, initial vector of design variables $\mathbf{x}_0$, objective (and constraint) functions, and desired RMSE for the estimators:
 1  Compute $\bar{a}_n$ and $\mathrm{Var}[\bar{a}_n]$ with $n$ large enough to meet desired RMSE at $\mathbf{x}_0$ using regular Monte Carlo simulation.
 2  Store in database $\mathbf{x}_0$ and the corresponding $\bar{a}_n$ and $\mathrm{Var}[\bar{a}_n]$ as $\hat{s}_C$ and $\mathrm{Var}[\hat{s}_C]$, respectively.
 3  Evaluate objective (and constraint) functions at $\mathbf{x}_0$ and $\bar{a}_n$ from Step 1.
 4  Optimization algorithm determines the next vector of design variables $\mathbf{x}_1$.
 5  Let $k = 1$.
 6  Compute $\ell = \arg\min_{\ell' < k} \|\mathbf{x}_k - \mathbf{x}_{\ell'}\|$
 7  Retrieve from database $\mathbf{x}_\ell$ and the corresponding $\hat{s}_C$ and $\mathrm{Var}[\hat{s}_C]$.
 8  Compute $\hat{s}_{A,p}$ and $\mathrm{Var}[\hat{s}_{A,p}]$ with $p$ large enough to meet desired RMSE using Algorithm 3.1.
 9  Store in database $\mathbf{x}_k$ and the corresponding $\hat{s}_{A,p}$ and $\mathrm{Var}[\hat{s}_{A,p}]$ as $\hat{s}_C$ and $\mathrm{Var}[\hat{s}_C]$, respectively.
10  Evaluate objective (and constraint) functions at $\mathbf{x}_k$ and $\hat{s}_{A,p}$ from Step 8.
11  Optimization algorithm determines the next vector of design variables $\mathbf{x}_{k+1}$.
12  If optimization convergence criteria is not met, set $k \leftarrow k + 1$ and return to Step 6; otherwise, stop.
```

3.2.4 Estimating Functions of Statistics

The information reuse estimator can be used to calculate statistics other than the mean, such as the variance, by redefining the samples as described in §2.2.3 for the multifidelity estimator. If the estimator is used to evaluate a function of one or more statistics, the error in the function estimator can be approximated by (2.5). Thus, similar to §2.2.4, we need to generalize the scalar information reuse estimator $\hat{s}_{A,p}$ in (3.2) to the vector case $\hat{\mathbf{s}}_{A,p}$.

Let $\mathbf{A}(\omega)$ and $\mathbf{C}(\omega)$ be $q \times 1$ random vectors and let $\boldsymbol{\Sigma}_A = \mathrm{Cov}\left[\mathbf{A}(\omega)\right]$, $\boldsymbol{\Sigma}_C = \mathrm{Cov}\left[\mathbf{C}(\omega)\right]$, and $\boldsymbol{\Sigma}_{AC} = \mathrm{Cov}\left[\mathbf{A}(\omega), \mathbf{C}(\omega)\right]$ be their $q \times q$ covariance matrices and cross-covariance matrix. Also, let $\boldsymbol{\gamma}$ be a $q \times q$ diagonal matrix whose elements are the q (optimal) control parameters for each of the q components of $\hat{\mathbf{s}}_{A,p}$. Then, the $q \times 1$ vector of information reuse estimators is

$$\hat{\mathbf{s}}_{A,p} = \bar{\mathbf{a}}_n + \boldsymbol{\gamma}(\hat{\mathbf{s}}_C - \bar{\mathbf{c}}_n)$$

and the $q \times q$ covariance matrix of the vector of information reuse estimator is

$$\begin{aligned} \mathrm{Cov}\left[\hat{\mathbf{s}}_{A,p}\right] &= \mathrm{Cov}\left[\bar{\mathbf{a}}_n\right] + \boldsymbol{\gamma}(\mathrm{Cov}\left[\hat{\mathbf{s}}_C\right] + \mathrm{Cov}\left[\bar{\mathbf{c}}_n\right])\boldsymbol{\gamma}^\top - \mathrm{Cov}\left[\bar{\mathbf{a}}_n, \bar{\mathbf{c}}_n\right]\boldsymbol{\gamma}^\top - \boldsymbol{\gamma}\,\mathrm{Cov}\left[\bar{\mathbf{c}}_n, \bar{\mathbf{a}}_n\right] \\ &= \frac{1}{n}\Big[\boldsymbol{\Sigma}_A + \boldsymbol{\gamma}(n\,\mathrm{Cov}\left[\hat{\mathbf{s}}_C\right] + \boldsymbol{\Sigma}_C)\boldsymbol{\gamma}^\top - \boldsymbol{\gamma}\boldsymbol{\Sigma}_{CA} - (\boldsymbol{\gamma}\boldsymbol{\Sigma}_{CA})^\top\Big]. \end{aligned}$$

The covariance matrix of the vector of information reuse estimators can then be used to compute the approximation of the mean square error of the function estimator in (2.5). This error estimate is useful for predicting the number of samples needed to control the objective function noise during optimization.

3.3 Combined Estimator

In the preceding sections, we assumed that we do not have a low-fidelity model and instead relied on evaluating the high-fidelity model at a different vector of design variables as a source of approximate information to reduce the computational cost of estimating statistics of the high-fidelity model output. We return to the situation where an inexpensive, low-fidelity model is available and discuss how to combine the multifidelity estimator and the information reuse estimator to further increase the efficiency of uncertainty propagation in the context of optimization under uncertainty.

Let k be the current optimization iteration and let $\ell < k$ be a past optimization iteration as described in §3.1.2. Given vectors of design variables $\mathbf{x}_k$ and $\mathbf{x}_\ell$, we define the following random variables:

- $A(\omega) = M_{\text{high}}(\mathbf{x}_k, \mathbf{U}(\omega))$ is the random output of the high-fidelity model at $\mathbf{x}_k$;
- $B(\omega) = M_{\text{low}}(\mathbf{x}_k, \mathbf{U}(\omega))$ is the random output of the low-fidelity model at $\mathbf{x}_k$;
- $C(\omega) = M_{\text{high}}(\mathbf{x}_\ell, \mathbf{U}(\omega))$ is the random output of the high-fidelity model at $\mathbf{x}_\ell$;
- $D(\omega) = M_{\text{low}}(\mathbf{x}_\ell, \mathbf{U}(\omega))$ is the random output of the low-fidelity model at $\mathbf{x}_\ell$.

The multifidelity estimators for the exact statistics $s_A = \mathbb{E}\left[A(\omega)\right]$ and $s_C = \mathbb{E}\left[C(\omega)\right]$, as discussed in §2.1.2, are

$$\begin{aligned}\hat{s}_A &= \bar{a}_n + \alpha\big(\bar{b}_m - \bar{b}_n\big),\\ \hat{s}_C &= \bar{c}_n + \beta\big(\bar{d}_m - \bar{d}_n\big),\end{aligned}$$

respectively, for control parameters $\alpha, \beta \in \mathbb{R}$ and $m \gg n$. We apply the information reuse estimator formulation (3.2), but replace the regular Monte Carlo estimators $\bar{a}_n$ and $\bar{c}_n$ with the above multifidelity estimators instead. Therefore, the combined estimator of s_A, denoted as $\tilde{s}_{A,p}$, is

$$\begin{aligned}\tilde{s}_{A,p} &= \hat{s}_A + \gamma(\tilde{s}_C - \hat{s}_C)\\ &= \big[\bar{a}_n + \alpha\big(\bar{b}_m - \bar{b}_n\big)\big] + \gamma\big[\bar{c}_n + \beta\big(\bar{d}_m - \bar{d}_n\big)\big] \qquad (3.8)\end{aligned}$$

for control parameter $\gamma \in \mathbb{R}$ and computational effort p to be defined below, where $\tilde{s}_C$ is the combined estimator at optimization iteration ℓ.

The computational effort for the combined estimator is $p = 2(n + m/w)$, where w is the (assumed known) ratio of the average computation time per high-fidelity model evaluation to the average computation time per low-fidelity model evaluation. If we were to follow the developments in §2.1.2 or §3.1.2, we would derive the expression for the combined estimator variance $\mathrm{Var}\left[\tilde{s}_{A,p}\right]$ and minimize it with respect to all parameters α, β, γ, and $r = m/n > 1$ for fixed computational budget p. Unfortunately, this minimization is complicated and there is no tractable analytical result as in the multifidelity estimator and the information reuse estimator. Therefore, we propose a suboptimal approach to reduce the variance of the combined estimator whereby we determine the parameters sequentially.

We first calculate the optimal (but suboptimal overall) α and β for the multifidelity estimators $\hat{s}_A$ and $\hat{s}_C$, respectively. From §2.1.2, we have

$$\alpha^* = \rho_{AB}\frac{\sigma_A}{\sigma_B}, \qquad r^*_{AB} = \sqrt{\frac{w\rho^2_{AB}}{1-\rho^2_{AB}}}$$

and

$$\beta^* = \rho_{CD}\frac{\sigma_C}{\sigma_D}, \qquad r^*_{CD} = \sqrt{\frac{w\rho^2_{CD}}{1-\rho^2_{CD}}}.$$

To be conservative in expending computational effort to evaluate the low-fidelity model, we choose the ratio of the number of low-fidelity model evaluations to the number of high-fidelity model evaluations as

$$r^* = \frac{m}{n} = \min\{r^*_{AB}, r^*_{CD}\}.$$

Using these choices for α^*, β^*, and r^*, the variances of the multifidelity estimators, based on (2.3), are

$$\mathrm{Var}\left[\hat{s}^*_A\right] = \left[1 - \left(1 - \frac{1}{r^*}\right)\rho^2_{AB}\right]\frac{\sigma^2_A}{n}$$

and

$$\mathrm{Var}\left[\hat{s}^*_C\right] = \left[1 - \left(1 - \frac{1}{r^*}\right)\rho^2_{CD}\right]\frac{\sigma^2_C}{n}.$$

Next, we need to determine γ. Since we already have $\mathrm{Var}\left[\hat{s}^*_A\right]$ and $\mathrm{Var}\left[\hat{s}^*_C\right]$, the variance of the combined estimator is

$$\mathrm{Var}\left[\tilde{s}_{A,p}\right] = \mathrm{Var}\left[\hat{s}_A^*\right] + \gamma(\mathrm{Var}\left[\tilde{s}_C\right] + \mathrm{Var}\left[\hat{s}_C^*\right]) - 2\gamma\,\mathrm{Cov}\left[\hat{s}_A^*, \hat{s}_C^*\right].$$

Minimizing the variance with respect to γ gives

$$\gamma^* = \frac{\mathrm{Cov}\left[\hat{s}_A^*, \hat{s}_C^*\right]}{\mathrm{Var}\left[\tilde{s}_C\right] + \mathrm{Var}\left[\hat{s}_C^*\right]}$$

and

$$\mathrm{MSE}\left[\tilde{s}_{A,p}^*\right] = \mathrm{Var}\left[\tilde{s}_{A,p}^*\right] = \mathrm{Var}\left[\hat{s}_A^*\right] - \frac{\mathrm{Cov}\left[\hat{s}_A^*, \hat{s}_C^*\right]^2}{\mathrm{Var}\left[\tilde{s}_C\right] + \mathrm{Var}\left[\hat{s}_C^*\right]}. \tag{3.9}$$

Lastly, in order to use (3.9), we derive the expression for the covariance of the multifidelity estimators:

$$\begin{aligned}
\mathrm{Cov}\left[\hat{s}_A^*, \hat{s}_C^*\right] =\ & \mathrm{Cov}\left[\bar{a}_n + \alpha^*\left(\bar{b}_m - \bar{b}_n\right), \bar{c}_n + \beta^*\left(\bar{d}_m - \bar{d}_n\right)\right] \\
=\ & \mathrm{Cov}\left[\bar{a}_n, \bar{c}_n\right] + \beta^*\,\mathrm{Cov}\left[\bar{a}_n, \bar{d}_m\right] - \beta^*\,\mathrm{Cov}\left[\bar{a}_n, \bar{d}_n\right] + \alpha^*\,\mathrm{Cov}\left[\bar{b}_m, \bar{c}_n\right] \\
& - \alpha^*\,\mathrm{Cov}\left[\bar{b}_n, \bar{c}_n\right] + \alpha^*\beta^*\,\mathrm{Cov}\left[\bar{b}_m, \bar{d}_m\right] - \alpha^*\beta^*\,\mathrm{Cov}\left[\bar{b}_m, \bar{d}_n\right] \\
& - \alpha^*\beta^*\,\mathrm{Cov}\left[\bar{b}_n, \bar{d}_m\right] + \alpha^*\beta^*\,\mathrm{Cov}\left[\bar{b}_n, \bar{d}_n\right] \\
=\ & \frac{1}{n}(\rho_{AC}\sigma_A\sigma_C - \beta^*\rho_{AD}\sigma_A\sigma_D - \alpha^*\rho_{BC}\sigma_B\sigma_C + \alpha^*\beta^*\rho_{BD}\sigma_B\sigma_D) \\
& - \frac{1}{m}(-\beta^*\rho_{AD}\sigma_A\sigma_D - \alpha^*\rho_{BC}\sigma_B\sigma_C + \alpha^*\beta^*\rho_{BD}\sigma_B\sigma_D) \\
=\ & \frac{1}{n}\left[\rho_{AC}\sigma_A\sigma_C + \left(1 - \frac{1}{r^*}\right)(\alpha^*\beta^*\rho_{BD}\sigma_B\sigma_D - \alpha^*\rho_{BC}\sigma_B\sigma_C - \beta^*\rho_{AD}\sigma_A\sigma_D)\right] \\
=\ & \left[\rho_{AC} + \left(1 - \frac{1}{r^*}\right)(\rho_{AB}\rho_{CD}\rho_{BD} - \rho_{AB}\rho_{BC} - \rho_{CD}\rho_{AD})\right]\frac{\sigma_A\sigma_C}{n}
\end{aligned}$$

As in §2.2.4 and §3.2.4, we also generalize the scalar combined estimator $\tilde{s}_{A,p}$ in (3.8) to the vector case $\tilde{\mathbf{s}}_{A,p}$. Let $\mathbf{A}(\omega)$, $\mathbf{B}(\omega)$, $\mathbf{C}(\omega)$, and $\mathbf{D}(\omega)$ be $q\times 1$ random vectors. Let $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ be $q \times q$ diagonal matrices whose elements are the q (suboptimal) control parameters for each of the q components of $\tilde{\mathbf{s}}_{A,p}$. Also, let $\hat{\mathbf{s}}_A$ and $\hat{\mathbf{s}}_C$ be the vector form of the multifidelity estimators $\hat{s}_A$ and $\hat{s}_C$, respectively and let $\mathrm{Cov}\left[\hat{\mathbf{s}}_A\right]$ and $\mathrm{Cov}\left[\hat{\mathbf{s}}_B\right]$ be their covariance matrices, respectively, as derived in §2.2.4. Then, the $q \times 1$ vector of combined estimators is

$$\tilde{\mathbf{s}}_{A,p} = \hat{\mathbf{s}}_A + \boldsymbol{\gamma}(\tilde{\mathbf{s}}_C - \hat{\mathbf{s}}_C)$$

and the $q \times q$ covariance matrix of the vector of combined estimators is

$$\mathrm{Cov}\,[\tilde{\mathbf{s}}_{A,p}] = \mathrm{Cov}\,[\hat{\mathbf{s}}_A] + \boldsymbol{\gamma}(\mathrm{Cov}\,[\tilde{\mathbf{s}}_C] + \mathrm{Cov}\,[\hat{\mathbf{s}}_C])\boldsymbol{\gamma}^\top - \boldsymbol{\gamma}\,\mathrm{Cov}\,[\hat{\mathbf{s}}_C, \hat{\mathbf{s}}_A] - (\boldsymbol{\gamma}\,\mathrm{Cov}\,[\hat{\mathbf{s}}_C, \hat{\mathbf{s}}_A])^\top.$$

Similar to the scalar case, we need the $q \times q$ cross-covariance matrix of the multifidelity estimators:

$$\begin{aligned}
\mathrm{Cov}\,[\hat{\mathbf{s}}_C, \hat{\mathbf{s}}_A] =&\ \mathrm{Cov}\left[\bar{\mathbf{a}}_n + \boldsymbol{\alpha}\big(\bar{\mathbf{b}}_m - \bar{\mathbf{b}}_n\big), \bar{\mathbf{c}}_n + \boldsymbol{\beta}\big(\bar{\mathbf{d}}_m - \bar{\mathbf{d}}_n\big)\right] \\
=&\ \frac{\boldsymbol{\Sigma}_{CA}}{n} + \frac{\boldsymbol{\Sigma}_{CB}}{m}\boldsymbol{\alpha}^\top - \frac{\boldsymbol{\Sigma}_{CB}}{n}\boldsymbol{\alpha}^\top + \boldsymbol{\beta}\frac{\boldsymbol{\Sigma}_{DA}}{m} - \boldsymbol{\beta}\frac{\boldsymbol{\Sigma}_{DA}}{n} \\
&+ \boldsymbol{\beta}\frac{\boldsymbol{\Sigma}_{DB}}{m}\boldsymbol{\alpha}^\top - \boldsymbol{\beta}\frac{\boldsymbol{\Sigma}_{DB}}{m}\boldsymbol{\alpha}^\top - \boldsymbol{\beta}\frac{\boldsymbol{\Sigma}_{DB}}{m}\boldsymbol{\alpha}^\top + \boldsymbol{\beta}\frac{\boldsymbol{\Sigma}_{DB}}{n}\boldsymbol{\alpha}^\top \\
=&\ \frac{1}{n}\Big[\boldsymbol{\Sigma}_{CA} + \Big(1 - \frac{n}{m}\Big)\big(\boldsymbol{\beta}\boldsymbol{\Sigma}_{DB}\boldsymbol{\alpha}^\top - \boldsymbol{\Sigma}_{CB}\boldsymbol{\alpha}^\top - \boldsymbol{\beta}\boldsymbol{\Sigma}_{DA}\big)\Big],
\end{aligned}$$

where $\boldsymbol{\Sigma}_{CA} = \mathrm{Cov}\,[\mathbf{C}(\omega), \mathbf{A}(\omega)]$, $\boldsymbol{\Sigma}_{DB} = \mathrm{Cov}\,[\mathbf{D}(\omega), \mathbf{B}(\omega)]$, and $\boldsymbol{\Sigma}_{DA} = \mathrm{Cov}\,[\mathbf{D}(\omega), \mathbf{A}(\omega)]$ are the cross-covariance matrices of the random vectors.

Finally, as in the information reuse estimator, we include a safeguard mechanism to prevent the combined estimator from requiring more computational effort than the multifidelity estimator. The procedure is the same as that described in 3.2.2, except that we fall back to the multifidelity estimator instead of to the regular Monte Carlo estimator.

3.4 Numerical Results

We revisit the acoustic horn robust optimization problem from §2.4.3. We compare the efficiency of the information reuse estimator against the regular Monte Carlo estimator in §3.4.1. In particular, we demonstrate the increase in the correlation coefficient ρ_{AC} between the random output of the model at $\mathbf{x}_k$ and the random output of the model at $\mathbf{x}_\ell$ as the distance between the vectors of design variables $\|\mathbf{x}_k - \mathbf{x}_\ell\|$ becomes shorter. Finally, we compare the combined estimator, the information reuse estimator, the multifidelity estimator, and the regular Monte Carlo estimator in §3.4.2.

3.4.1 Acoustic Horn Robust Optimization Revisited

We return to the 2-D acoustic horn governed by the non-dimensional complex Helmholtz equation described in §2.4.2. The model we consider is the reduced basis model for the reflection coefficient s with $N = 30$ basis functions. The three random input parameters are the wave number, upper horn wall impedance, and lower horn wall impedance with distributions listed in Table 2.2. The design variables are $\mathbf{b} = [b_1 \cdots b_6]^\top$ representing the half-widths of the horn flare as shown in Figure 2-6. The initial values of the design variables (corresponding to the straight flare), their lower bounds, and their upper bounds are listed in Table 2.3. The minimization of the horn reflection coefficient is formulated as

$$\min_{\mathbf{b}_L \leq \mathbf{b} \leq \mathbf{b}_U} f(\mathbf{b}) = \mathbb{E}\left[s(\mathbf{b}, \omega)\right] + 3\sqrt{\mathrm{Var}\left[s(\mathbf{b}, \omega)\right]}.$$

We employ the bound optimization by quadratic approximation algorithm (BOBYQA) [42] developed for bound-constrained optimization problems without analytical derivatives. The algorithm constructs an underdetermined quadratic interpolation model of the objective function based on the least-Frobenius-norm update of the Hessian. The quadratic model is used in a trust-region subproblem to generate the vectors of design variables. The optimization is conducted with the objective function evaluated using the regular Monte Carlo estimator and with the objective function evaluated using the information reuse estimator. In both cases, the tolerance on the RMSE of the $f(\mathbf{b})$ estimator is fixed at 1×10^{-3}. Three trials were run for each case and the convergence of the objective as a function of the cumulative computational effort is shown in Figure 3-2, where the computational effort is the number of high-fidelity model evaluations. It can be seen that using the information reuse estimator provides cumulative computational savings over the course of the optimization and we locate

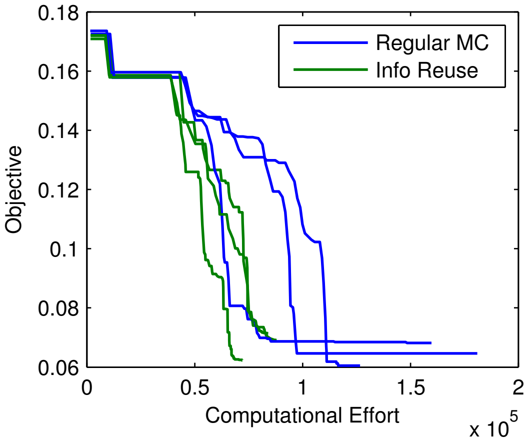
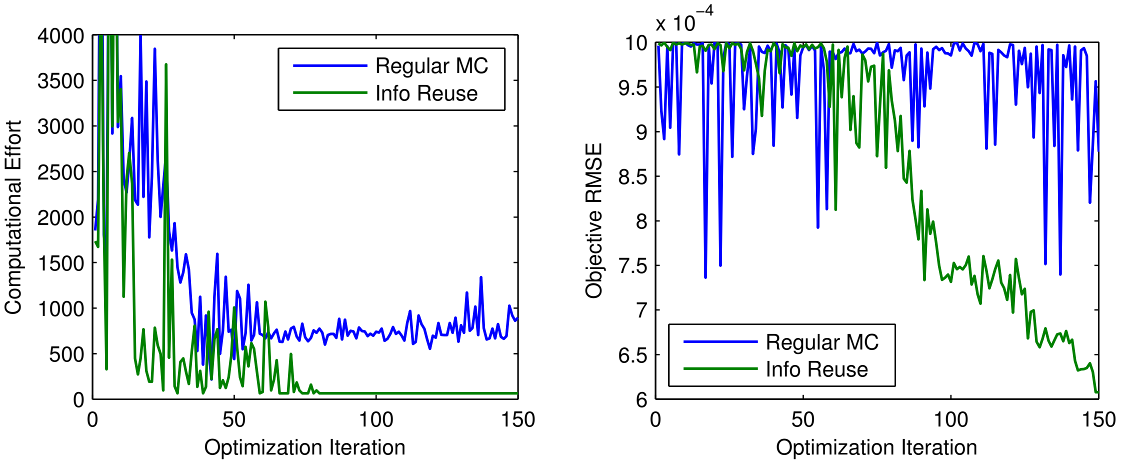


Figure 3-2: Comparison of convergence histories for the robust horn optimization using the regular Monte Carlo estimator and the information reuse estimator. The optimization algorithm is BOBYQA.

the optimum using about 50% less computational effort.

We examine the computational cost in more detail by plotting the computational effort used to compute the regular Monte Carlo estimator and the computational effort used to compute the information reuse estimator at each optimization iteration in Figure 3-3a. We see that, while both estimators require about the same computational effort during the first few optimization iterations, the information reuse estimator requires significantly less computational effort at subsequent optimization iterations, eventually settling at $p = 2n_{\text{init}}$. In fact, at later optimization iterations, the computational effort required is less than $2n_{\text{init}}$. Since we maintain at least $n = n_{\text{init}}$ samples to calculate the parameters in (3.5) needed by the information reuse estimators, the RMSE of the objective actually decreases and becomes smaller than the specified tolerance of 1×10^{-3} as shown in Figure 3-3b.

The reduction in computational effort corresponds to the high correlation coefficient $\hat{\rho}_{AC}$ shown in Figure 3-4a due to the optimizer taking smaller and smaller steps at the later optimization iterations as it refines the optimal vector of design variables. As discussed in §3.1.1, the correlation between the random model output at a vector of design variables and the random model output at another vector of design variables tend to increase as the distance between the two vectors of design variables decreases. On the other hand, when the correlation coefficient $\hat{\rho}_{AC}$ is not high enough, it may



(a) Computational Effort p

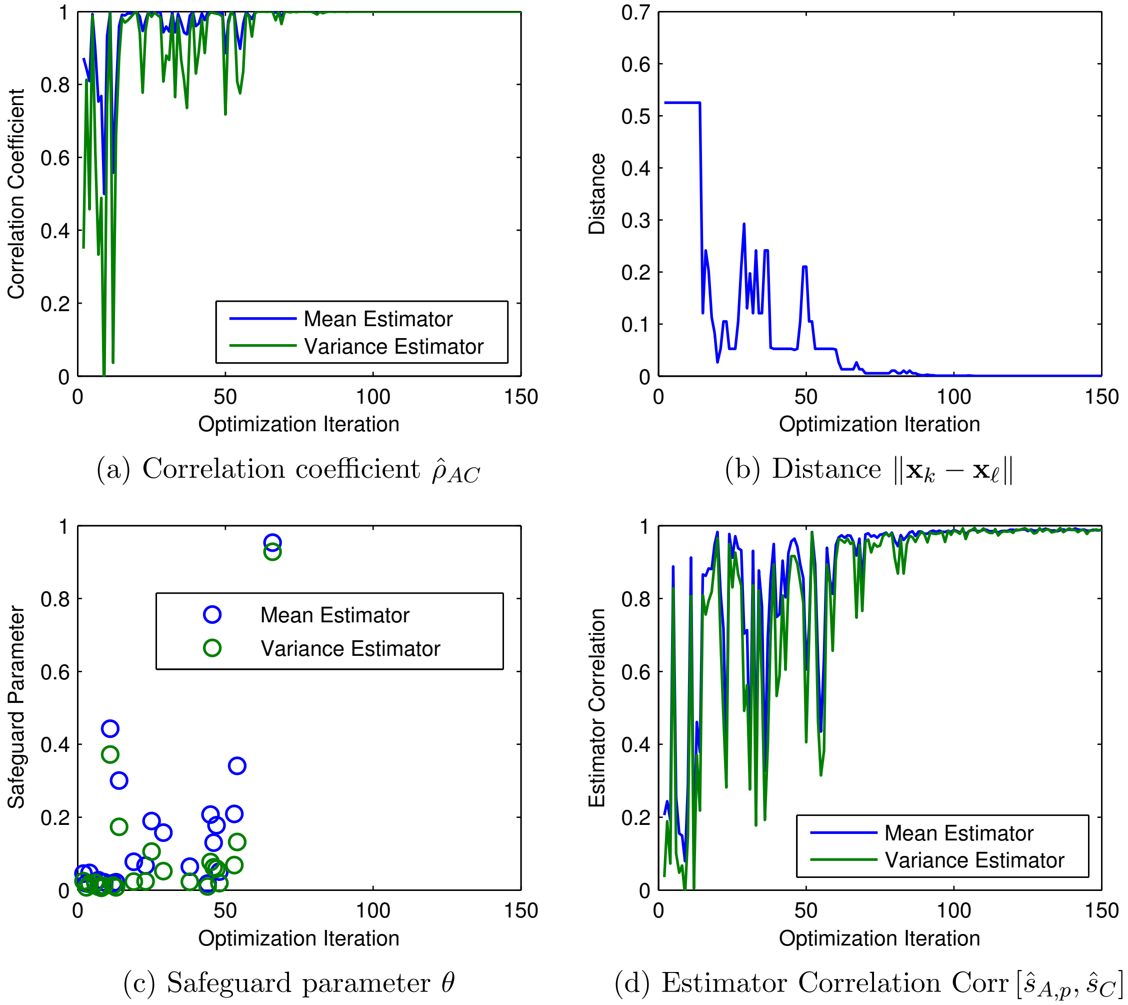
(b) Objective RMSE

Figure 3-3: Computational effort per optimization iteration and the root mean square error of the objective versus optimization iteration for the robust horn optimization example.

be necessary to safeguard the information reuse estimator to prevent it from requiring more computational effort than the regular Monte Carlo estimator. We plot the safeguard parameter θ in Figure 3-4c and show that this has indeed occurred during the first few optimization iterations.

In Figure 3-4d, we plot the correlation coefficient between the estimators $\hat{s}_{A,p}$ and $\hat{s}_C$ as derived in (3.4). It shows that the errors in the estimators at the later optimization iterations become more and more correlated with each other. As discussed in §3.1.3, this correlation has the effect of smoothing the noise in the objective function at the cost of introducing a bias. This may help prevent the derivative-free optimization algorithm from getting stuck in the noise of the objective function by causing it to behave more like the sample average approximation method. Furthermore, as shown in Figure 3-3b, the estimator variances (not conditioned on the previous estimators) decreases at the later optimization iterations. Therefore, unlike the sample average approximation method, the bias can decrease as the optimization algorithm progresses.

Finally, we verify our theoretical RMSE for the information reuse estimator from (3.3) by comparing it with the empirical RMSE. We fix the sequence of vectors of design variables generated by the optimization and apply Algorithm 3.2 to calculate



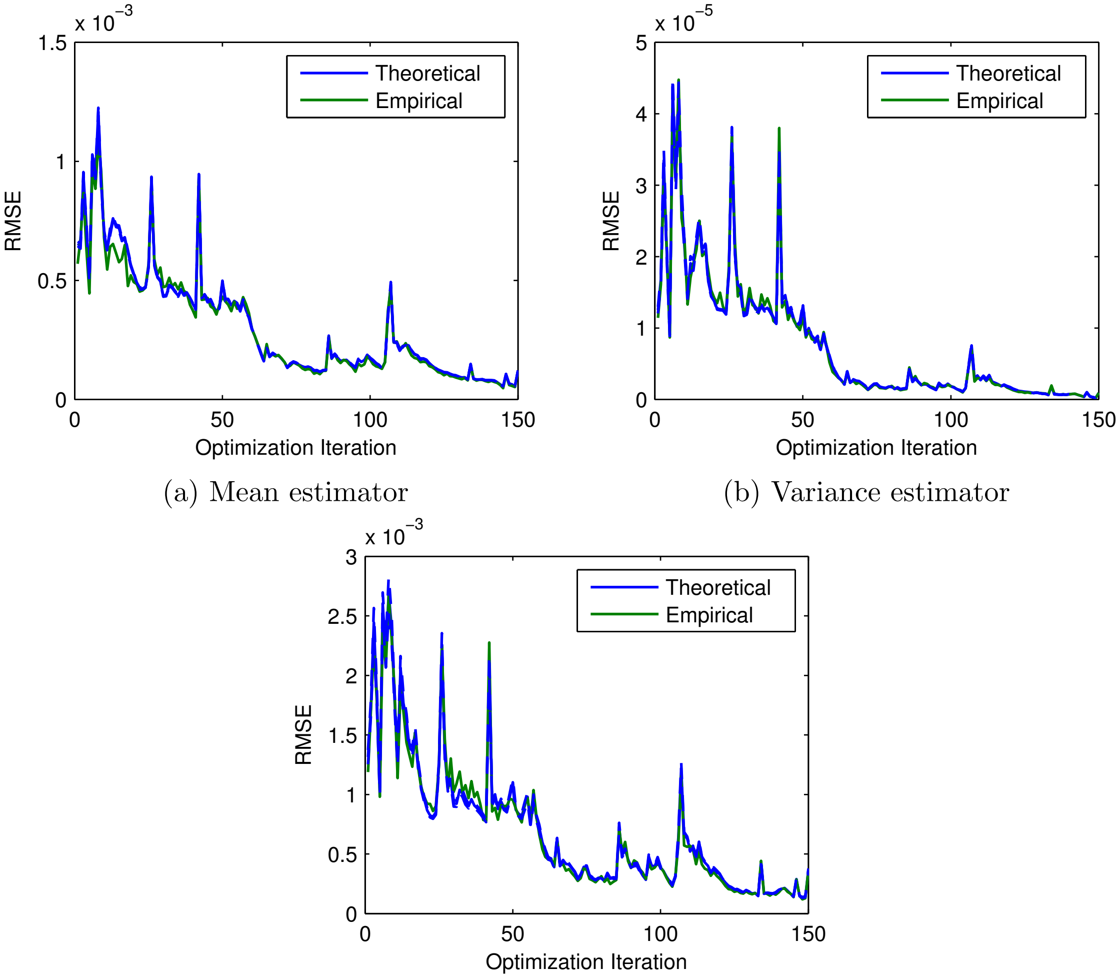
(a) Correlation coefficient $\hat{\rho}_{AC}$

(b) Distance $\|\mathbf{x}_k - \mathbf{x}_\ell\|$

(c) Safeguard parameter θ

(d) Estimator Correlation $\mathrm{Corr}\left[\hat{s}_{A,p}, \hat{s}_C\right]$

Figure 3-4: The model output correlation coefficient, the distance between vectors of design variables, the safeguard parameter, and the estimator correlation coefficient for the information reuse estimator versus optimization iteration for the robust horn optimization example.



(a) Mean estimator

(b) Variance estimator

(c) $f = \mathbb{E}\left[s\right] + 3\sqrt{\mathrm{Var}\left[s\right]}$ estimator

Figure 3-5: Comparison of the theoretical root mean square errors to the empirical root mean square errors of the information reuse estimators for the robust horn optimization example.

the information reuse estimators on this sequence of vectors of design variables, expending $p = 1000$ computational effort on every vector of design variables. We repeat this with new realizations of the random inputs 100 times to calculate the empirical RMSE of the information reuse estimators. The results are shown in Figure 3-5 and show good agreement.

3.4.2 Multifidelity Acoustic Horn Robust Optimization

For the last example of the acoustic horn robust optimization problem, we compare the regular Monte Carlo estimator, the information reuse estimator, the multifidelity estimator, and the combined estimator. The problem setup is the same as that in §3.4.1; however, we now consider the high-fidelity model to be the finite element model of the Helmholtz equation and the low-fidelity model to be the reduced basis model with $N = 30$ basis functions. Figure 3-6 shows the convergence of the objective with respect to the cumulative computational effort, where the computational effort is the number of high-fidelity model evaluations for the regular Monte Carlo estimator and the information reuse estimator and is the equivalent number of high-fidelity model evaluations for the multifidelity estimator and the combined estimator. It can be seen that the combined estimator requires significantly less computational effort than the regular Monte Carlo estimator. For this problem, the multifidelity estimator is already quite efficient and so the combined estimator does not provide much additional benefit. Nevertheless, Figure 3-7 shows that the combined estimator indeed combines the benefits of both the multifidelity estimator and the information reuse estimator—at the first few optimization iterations, when the optimizer takes relative large steps in the design space, the combined estimator takes about the same computational effort as the multifidelity estimator; at the last few optimization iterations, when the optimizer takes relative small steps in the design space, the combined estimator takes only $p = 2n_{\text{init}}$ computational effort as in the case of the information reuse estimator. Overall the information reuse estimator provided about 60% computational savings, the multifidelity estimator provided about 75% computational savings, and the combined estimator provided about 90% computational savings, as shown in Table 3.1.

Table 3.2 presents the four slightly different optimal solutions obtained using each of the four estimators and shows the challenge in solving noisy optimization problems using derivative-free optimization algorithms. While the RMSE of 1×10^{-3} is two orders of magnitude less than the initial objective value, it is only one order of

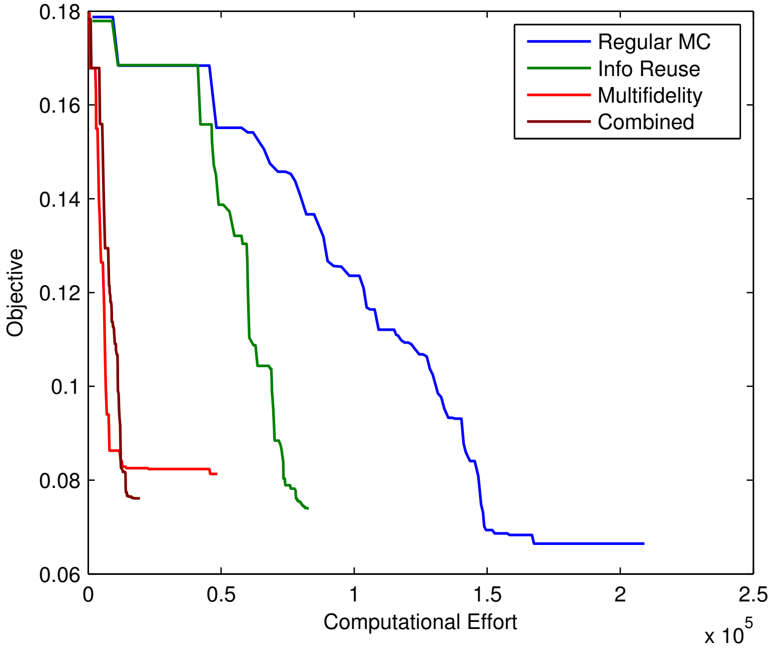


Figure 3-6: Comparison of convergence histories for the robust horn optimization using the regular Monte Carlo estimator, the information reuse estimator, the multifidelity estimator, and the combined estimator. The optimization algorithm is BOBYQA.

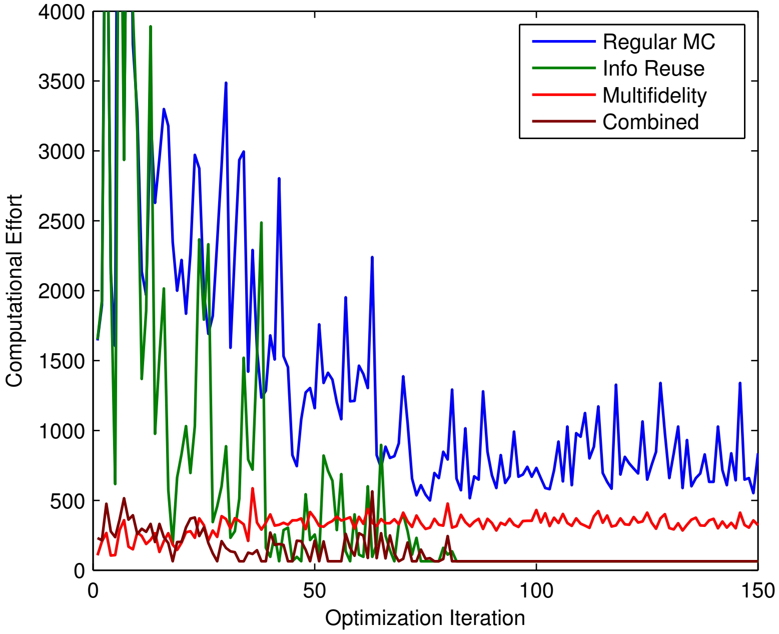


Figure 3-7: Computational effort per optimization iteration versus optimization iteration for the robust horn optimization example.

Table 3.1: Comparison of the total computational efforts for the robust horn optimization using the regular Monte Carlo estimator, the information reuse estimator, the multifidelity estimator, and the combined estimator. The optimization algorithm is BOBYQA.

	Total Computational Effort
Regular Monte Carlo estimator	209,096
Information reuse estimator	82,928
Multifidelity estimator	48,595
Combined estimator	19,449

Table 3.2: Comparison of the final design variables for the robust horn optimization using the regular Monte Carlo estimator, the information reuse estimator, the multifidelity estimator, and the combined estimator.

Regular MC	Info Reuse	Multifidelity	Combined
0.679	0.679	0.679	0.679
1.06	1.08	1.10	1.06
1.75	1.67	1.66	1.69
1.96	1.86	1.89	1.84
2.26	2.11	2.11	2.14
2.46	2.46	2.52	2.48

magnitude less than the final objective value. As discussed in §2.3.3, derivative-free optimization methods are not guaranteed to converge to the true optimal solution in the presence of noise and may terminate prematurely. If computational resources allow, one may perform multiple runs of the optimization to obtain a spread of the final objective values, similar to what is sometimes done for the sample average approximation method. For the acoustic horn problem, the spread of final objective values for the different estimators would overlap, as can be seen to certain degree in Figure 2-8 and Figure 3-2.

3.5 Chapter Summary

In this chapter, we presented an approach to estimate statistics of the output of an expensive high-fidelity model within an outer optimization loop by reusing the estimates of the statistics from previous optimization iterations. We showed that as the distance between the vectors of design variables become shorter, the correlation between the random output of the model at the two vectors of design variables increases. This means that the estimates of the statistics from the previous optimization iteration are informative and can be used to reduce the computational cost of estimating the statistics at the current optimization iteration. Numerical results for the acoustic horn robust optimization example demonstrated 90% computational savings.

In the next chapter, we demonstrate that the estimators developed in Chapter 2 and Chapter 3 enable design under uncertainty for practical aerospace problems.

Chapter 4

Applications in Aerostructural Optimization Under Uncertainty

In this chapter, we apply the methods developed in Chapter 2 and Chapter 3 to two demonstrative applications that are relevant to the aerospace industry. Optimization under uncertainty for industrial problems can be challenging for at least two reasons. First, numerical models that capture high-fidelity physics of the engineering system can be computationally expensive to evaluate. The solvers may also be closed inside a "black-box", preventing the use of intrusive methods that require the knowledge of the governing equations. In particular, the black-box models may not provide gradients of the outputs that are needed by efficient optimization methods. It may even fail to evaluate for certain inputs (unknown a priori) due to hidden constraints. Second, it may be difficult to fully characterize all of the uncertainties associated with the numerical model for a particular problem due to the lack of expertise and/or resources. The latter issue is beyond the scope of this work and the forms of the uncertainties for the following application problems have been provided. Instead, we focus on reducing the computational cost of optimization under uncertainty using the multifidelity estimator, information reuse estimator, and combined estimator. For the first application in §4.1, we consider the conceptual design of an aircraft with uncertainties arising from simplified physics in the model and projections about future technology improvements. For the second application in §4.2, we consider a large-scale detailed wing design problem with uncertainties in the structural properties and flight conditions.

4.1 Aircraft Conceptual Design Under Uncertainty

The first application is the conceptual design of an advanced aircraft for the 2035 entry-to-service time frame. We perform a robust optimization using both the regular Monte Carlo estimator and the information reuse estimator and compare their computational costs. We also demonstrate how the information reuse estimator can be used efficiently to study the risk-performance trade-off that requires solving multiple robust optimization problems.

4.1.1 Problem Setup

The D8 family of aircraft was developed at MIT, Aurora Flight Sciences, and Pratt and Whitney as part of a NASA-sponsored project to identify enabling technologies and innovative configurations that will allow a subsonic commercial transport aircraft to meet the N+3 goals by the year 2035 [16]. The goals include 70% reduction in fuel burn, 71 dB reduction in effective perceived noise level, and 75% reduction in landing and take-off NOx relative to the current generation of Boeing 737-800 aircraft. The D8 aircraft concept, rendered in Figure 4-1, is in the same class as the Boeing 737-800, carrying 180 passengers over a range of 3000 nautical miles. The highlights of its features include engines that ingest the fuselage boundary layer, wide lift-generating "double-bubble" fuselage as illustrated in Figure 4-2, engine noise shielding from the fuselage and vertical tails, new composite materials, advanced engine thermodynamic cycle, and active load alleviation [16]. Many of these advanced technologies are still in development and their expected benefits can only be predicted based on a combination of historical projections, preliminary simulations, and industry expert opinions. Therefore, there are uncertainties about the effects of each of these technologies that can impact the overall performance of the aircraft. The purpose of this application problem is to demonstrate an approach to account for these uncertainties and hedge

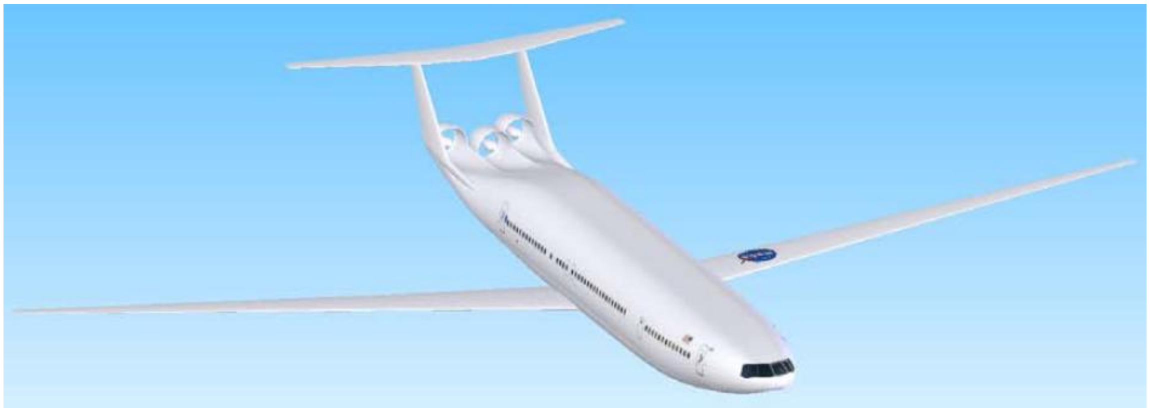


Figure 4-1: Rendering of the D8 aircraft [16, Figure 1].

against the risks that they represent during the design process. We demonstrate how the computational cost of achieving this can be reduced by using the information reuse estimator developed in Chapter 3.

The numerical model for this problem is the Transport Aircraft System OPTimization (TASOPT) software [17]. It is a multidisciplinary tool for aircraft sizing and mission performance analysis with modules in aerodynamics, structural loads and weights, engine cycle analysis, and trajectory simulation. In order to support the design of the D8 family of aircraft with a step-change in technology, TASOPT is developed from first principles using low-order physics rather than from historical correlations. This allows TASOPT to capture improvements beyond incremental change from past aircraft that are needed to meet the ambitious N+3 goals. For this application problem, we perform a robust optimization on the D8.6 aircraft subject to randomness in the TASOPT parameters that represents the uncertainties in the advanced technologies. Therefore, the internal optimization capability of TASOPT is disabled and, given a set of design variables and a realization of the random parameters, it is run only in forward mode to size the aircraft and calculate mission performance.

We consider the D8.6 aircraft for the robust optimization problem. There are eight design variables ($\mathbf{x}$) representing the wing geometry and cruise conditions. Their initial values ($\mathbf{x}_0$) and bounds ($\mathbf{x}_L$ and $\mathbf{x}_U$) are listed in Table 4.1. The 19 random inputs ($\mathbf{U}(\omega)$) are parameters in the TASOPT low-order physics models whose values

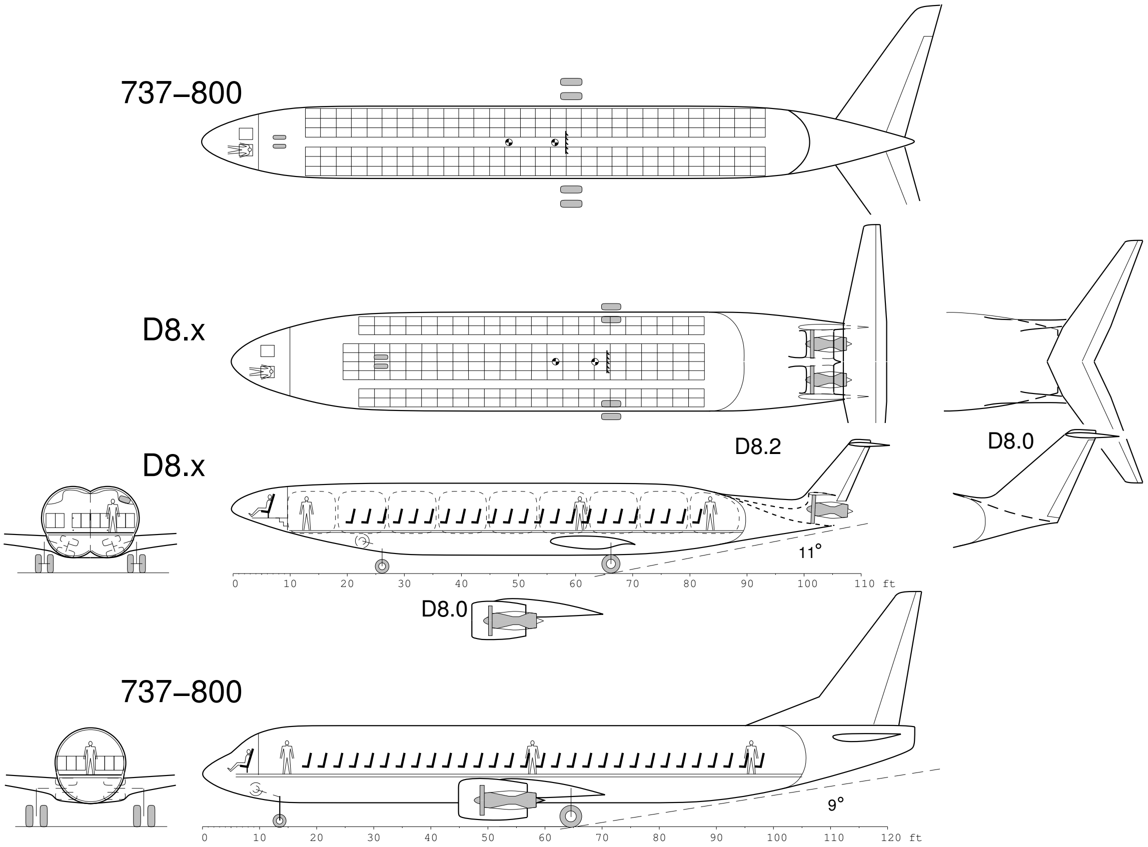

737–800
D8.x
D8.x
D8.2
D8.0
11°
0 10 20 30 40 50 60 70 80 90 100 110 ft
D8.0
737–800
9°
0 10 20 30 40 50 60 70 80 90 100 110 120 ft

Figure 4-2: Comparison of the D8 aircraft and the Boeing 737-800 aircraft [10, Figure 22].

are chosen to reflect the predicted improvements in technologies. The uncertainties in the predictions are represented by triangular random distributions with the distribution parameters listed in Table 4.2.[1] The objective of the robust optimization problem is the mean fuel burn for the mission, i.e. carrying 180 passengers over 3000 nautical miles. It is expressed in terms of the payload fuel energy intensity (PFEI), which is the energy of the fuel used per unit payload and unit range [kJ/(kg km)]. Given the design variables $\mathbf{x}$ and a realization of the random inputs $\mathbf{u}$, TASOPT solves a set of governing equations to generate a feasible aircraft—that is, sizing and positional degrees of freedom of the aircraft components are taken up interally to satisfy constraints in loads, stability, trajectory, etc. However, there are four additional performance requirements of interest on the aircraft that are treated externally as constraints for the optimization problem. For the deterministic problem, these constraints are:

$$\begin{aligned}
&\text{req}_1(\mathbf{x},\mathbf{u}) = \text{balanced field length} - 4995 \text{ ft} \leq 0\\
&\text{req}_2(\mathbf{x},\mathbf{u}) = \text{required fuel volume} - 90\% \text{ of available internal wing volume} \leq 0\\
&\text{req}_3(\mathbf{x},\mathbf{u}) = \text{span length} - 177.5 \text{ ft} \leq 0\\
&\text{req}_4(\mathbf{x},\mathbf{u}) = 0.015 - \text{top-of-climb gradient} \leq 0.
\end{aligned}$$

In the presence of random inputs, all of the performance requirements are formulated as mean + λ standard deviations ≤ 0 constraints with $\lambda = 1$. Therefore, the robust

[1] The triangular distributions represent reasonable assumptions about the uncertainties. They are not the result of a rigorous elicitation process, which is beyond the scope of this work.

Table 4.1: Initial values, lower bounds, upper bounds, and optimal values of the 8 D8.6 aircraft design variables.

Variable	Initial Value	Lower Bound	Upper Bound	Optimal Value
Cruise lift coefficient	0.71092	0.3	1.0	0.71248
Wing aspect ratio	17.07	5	30	16.668
Wing sweep [deg]	17.757	10	30	17.451
Wing thickness at root	0.13441	0.08	0.20	0.13331
Wing thickness at break and tip	0.10079	0.08	0.20	0.10154
Wing cruise lift distribution fraction at break	1.136	0.5	1.5	1.1341
Wing cruise lift distribution fraction at tip	1.2645	0.5	1.5	1.2672
Start of cruise altitude [m]	11,784	10,000	13,000	11,785

optimization problem is

$$\begin{aligned}
\min_{\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U} \quad & \mathbb{E}\left[\mathrm{PFEI}(\mathbf{x}, \mathbf{U}(\omega))\right] \\
\text{s.t.} \quad & \mathbb{E}\left[\mathrm{req}_1(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{req}_1(\mathbf{x}, \mathbf{U}(\omega))\right]} \leq 0 \\
& \mathbb{E}\left[\mathrm{req}_2(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{req}_2(\mathbf{x}, \mathbf{U}(\omega))\right]} \leq 0 \\
& \mathbb{E}\left[\mathrm{req}_3(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{req}_3(\mathbf{x}, \mathbf{U}(\omega))\right]} \leq 0 \\
& \mathbb{E}\left[\mathrm{req}_4(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{req}_4(\mathbf{x}, \mathbf{U}(\omega))\right]} \leq 0.
\end{aligned}$$

4.1.2 Numerical Results

We employ the constrained optimization by linear approximation algorithm (COBYLA) [41] developed for constrained optimization problems without analytical derivatives. The algorithm constructs linear interpolation models of the objective and constraint functions using evaluations on a simplex. The vectors of design variables are obtained either by solving a linear programming subproblem using the interpolation models or by improving the geometry of the simplex. The optimization is conducted with the objective and constraint functions evaluated using the regular Monte Carlo estimator and with the objective and constraint functions evaluated using the information reuse estimator. In both cases, the tolerance on the RMSE of the objective function estimator is fixed at 1×10^{-3} while the tolerance on the RMSE of the con-

Table 4.2: Triangular distributions of the 19 random inputs for the D8.6 aircraft.

Random Variable	Lower Limit	Mode	Upper Limit
Vertical load factor for wing bending	2.3	2.5	3.0
Secondary wing components weight fraction	0.49	0.54	0.59
Secondary engine components weight fraction	0.0	0.1	0.2
Material yield stress multiplier	0.8	1.0	1.2
Material density [lb/in^3]	0.0504	0.0560	0.0616
Wing excrescence drag factor	1.019	1.025	1.038
Tail excrescence drag factor	1.019	1.025	1.038
Fuselage excrescence drag factor	1.03	1.04	1.08
Fuselage boundary layer ingestion fraction	0.2	0.4	0.6
Turbine blade metal temperature [K]	1450	1500	1550
Turbine cooling Stanton number	0.050	0.065	0.080
Turbine cooling heat transfer efficiency	0.6	0.7	0.8
Turbine cooling film effectiveness factor	0.3	0.4	0.5
Engine overall pressure ratio	45	50	52
Fan efficiency	0.930	0.945	0.950
Low pressure compressor efficiency	0.89	0.93	0.94
High pressure compressor efficiency	0.88	0.90	0.93
High pressure turbine efficiency	0.880	0.925	0.940
Low pressure turbine efficiency	0.91	0.93	0.95

straint function estimators is fixed at 5×10^{-4}. The convergence of the objective as a function of the cumulative computational effort is shown in Figure 4-3, where the computational effort is the number of high-fidelity model evaluations. It can be seen that using the information reuse estimator provides cumulative computational savings over the course of the optimization and we locate the optimum using about 90% less computational effort.

In Figure 4-4, we plot the computational effort required to compute the regular Monte Carlo estimator and the information reuse estimator at each optimization iteration. After the first few optimization iterations, the information reuse estimator requires significantly less computational effort than the regular Monte Carlo estimator to meet the required RMSE tolerance because the database of past optimization iterations has built up sufficiently to provide good candidates for the auxiliary random variable. This is particularly advantageous when there is a need to refine the optimal solution to high accuracy (assuming the RMSE's of the objective and constraint function estimators are low enough to do so) because the additional optimization iterations needed would require relatively little computational effort using the information reuse

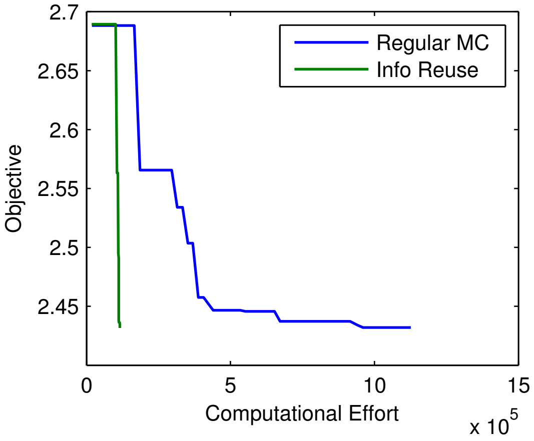


Figure 4-3: Comparison of convergence histories for the D8.6 aircraft robust optimization using the regular Monte Carlo estimator and the information reuse estimator. The objective is the mean PFEI in kJ/(kg km). The optimization algorithm is COBYLA.

estimator.

4.1.3 Risk-Performance Trade-off Studies

Because of the randomness representing the uncertainties in the aircraft design, there is always a risk that the realized aircraft will not satisfy one or more of the performance requirements. Therefore, it is useful to study the trade-off between this risk and the expected fuel burn of the optimal aircraft.[2] For the robust optimization problem in §4.1.2, $\lambda = 1$ is chosen somewhat arbitrarily. Here, we let λ vary from 0 to 3 and re-solve the robust optimization problem for each value of λ in order to generate a Pareto front of the risk, defined as the probability of not satisfying the performance requirements, versus the expected fuel burn of the optimal aircraft.

Generating the Pareto front is computationally expensive even with the savings provided by the information reuse estimator because many robust optimization problems have to be solved. Fortunately, each optimization problem need not be solved from scratch. As discussed in §4.1.2, the first few optimization iterations using the information reuse estimator require the most computational effort because the database

[2] The risk may be defined in other ways depending on the context, such as the standard deviation of the objective.

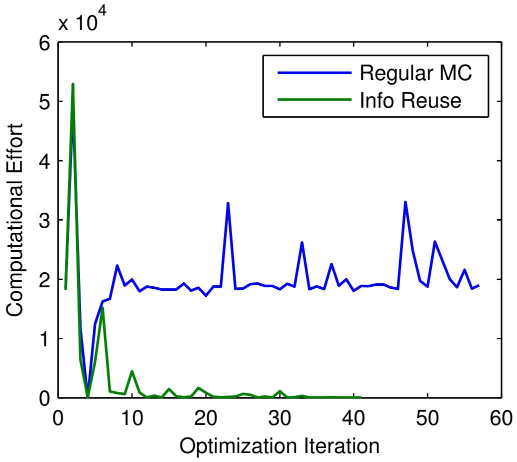


Figure 4-4: Computational effort per optimization iteration versus optimization iteration for the D8.6 aircraft robust optimization.

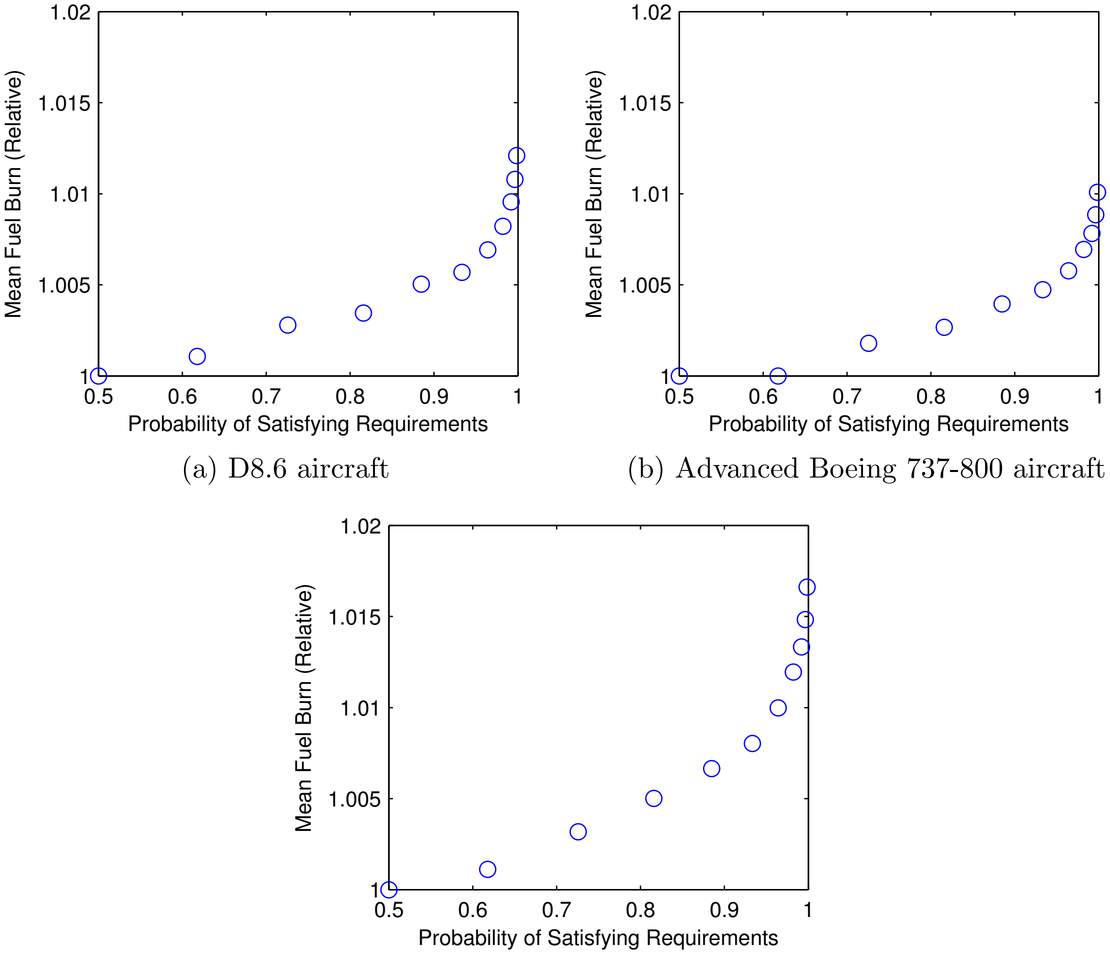
of past optimization iterations has yet to be built up. To reduce this computational cost, we reuse the database from the previous robust optimization in the next robust optimization. This is possible because the only difference between the multiple robust optimization problems is the value of λ; the same statistics, i.e., the mean and the variance of the TASOPT outputs, are calculated for all vectors of design variables in all of the robust optimization problems. The result is that there are good candidates for the auxiliary random variable starting at the first optimization iteration (except for the first robust optimization problem) and therefore the information reuse estimator requires relatively little computational effort.

The Pareto front for the D8.6 aircraft is plotted in Figure 4-5a. We see that, while it is necessary to sacrifice expected fuel burn to reduce the risk of not satisfying the performance requirements, the increase in expected fuel burn is not large, demonstrating that the D8.6 is relatively robust to the uncertainties. In Figure 4-6a, we show the trend in the wing aspect ratio along the Pareto front. It can be seen that the reduction in the wing aspect ratio is a major driver for the increase in expected fuel burn; other design variables do not show such a clear trend. If we solved the deterministic optimization problem with the input parameters fixed at the mode of the triangular distributions and then introduce the uncertainties to the deterministic optimal design, we find that its mean fuel burn is 0.999 on the relative scale in Figure 4-5a but its probability of satisfying the performance requirements is only about 30%—the span length requirement being the critical constraint. This demonstrates the importance in accounting for uncertainties during the design process.

As a comparison, we also generate similar Pareto fronts for the Boeing 737-800 aircraft and the Boeing 777-300ER aircraft. The Boeing 737-800 aircraft is in the same class as the D8.6 aircraft with 180 passengers and 3000 nautical-mile-range while the Boeing 777-300ER aircraft is larger with 450 passengers and 6000 nautical-mile-range. We assume that both Boeing aircraft are outfitted with advanced technologies along with the uncertainties similar to the D8.6 aircraft. However, they do not have fuselage boundary layer ingestion because it is incompatible with their configurations. The results are shown in Figure 4-5b and Figure 4-5c. It can be seen that the Pareto front for the Boeing 737-800 aircraft is similar to that of the D8.6 aircraft, indicating that it is also relatively robust to the uncertainties. However, the Boeing 777-300ER needs to trade off more expected fuel burn to satisfy the performance requirements. For this study, the larger aircraft is more sensitive to uncertainties in the technologies that underlie their performance.

4.2 Robust Multifidelity Wing Optimization

The second application is a large-scale aerostructural wing design problem that involves a detailed discretization of the wing geometry and internal structures. We demonstrate that a coarse discretization can be an effective low-fidelity model by comparing the computational efforts required by the information reuse estimator and by the combined estimator to solve the robust optimization problem.

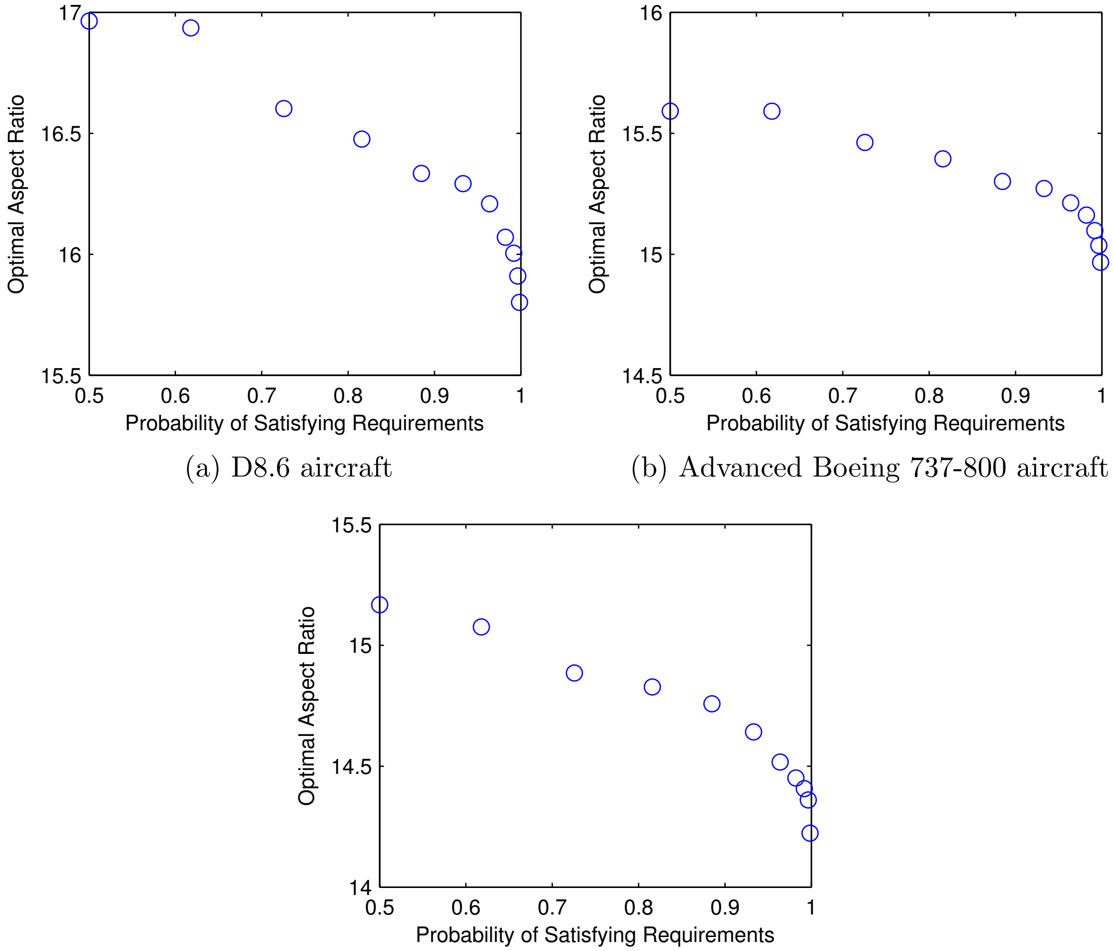


(a) D8.6 aircraft

(b) Advanced Boeing 737-800 aircraft

(c) Advanced Boeing 777-300ER aircraft

Figure 4-5: Pareto fronts of mean fuel burn versus probability of satisfying requirements.



(a) D8.6 aircraft

(b) Advanced Boeing 737-800 aircraft

(c) Advanced Boeing 777-300ER aircraft

Figure 4-6: Trends of the optimal wing aspect ratio along the Pareto front.

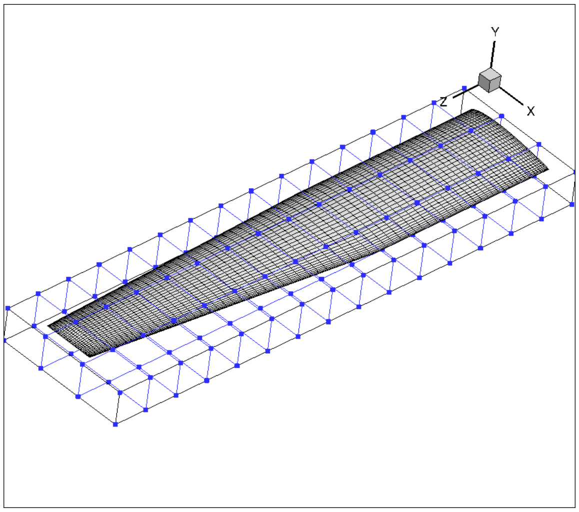


Figure 4-7: Geometry and the free-form deformation control points of the Bombardier Q400 wing [34, Figure 6-8].

4.2.1 Problem Setup

We consider the robust optimization of a wing whose geometry is loosely based on the Bombardier Q400 aircraft. The numerical model is a coupled aerostructural solver using Tripan for the aerodynamic analysis and the Toolkit for the Analysis of Composite Structures (TACS) for the structural analysis [24]. Tripan is an unstructured panel method using source and doublet singularities for inviscid, incompressible, external lifting flow. TACS is a second-order finite element method using linear or nonlinear strain relationships for thin-walled structures. The aerostructural solver is parallelized at both the coupling level and at the individual analysis level to take advantage of multiple processors. A CAD-free approach based on free-form deformations [27] is used to parameterize the wing geometry. The wing and the free-form deformation control points are shown in Figure 4-7 and some data for the initial wing geometry are listed in Table 4.3.

In the deterministic optimization problem [24, 34], the drag of the wing is minimized subject to lift equals weight for 1-g cruise constraint, four Kreisselmeier-Steinhauser (KS) stress constraints for the ribs, the spars, the upper surface skin, and the lower surface skin, and 72 constraints on the magnitude of the change in thicknesses of spars and skins along the span for a total of 77 constraints. The design

Table 4.3: Geometry data of the Bombardier Q400 wing.

Planform Area [m^2]	68.44
Wing span [m]	28.40
Aspect ratio	11.78
Taper ratio	0.54
Taper break [% of span]	40
Airfoil stack	NACA 2412
Twist	None
Number of ribs	20
Forward spar location [% of chord]	15
Aft spar location [% of chord]	50

variables include the angle of attack, eight wing twist angles along the span, and 19 thicknesses of each of the forward spar, the aft spar, the ribs, the upper surface skin, and the lower surface skin for a total of 104 variables.[3] Since only the wing is optimized, any savings in wing weight is subtracted from the total aircraft weight for the lift constraint.

We introduce uncertainties into the cruise condition and structural properties using triangular distributions listed in Table 4.4. However, the deterministic optimization problem cannot be translated into a robust optimization problem in a straightforward manner because it is not clear how to apply the equality constraint of lift equals weight when the lift and the weight are random outputs. To resolve this issue, we wrap the secant root-finding method around the aerostructural solver to find the angle of attack that satisfies the lift constraint. This effectively moves the lift constraint out of the optimization problem and into the numerical model and eliminates the angle of attack from the design variables. The interpretation is that each set of design variables (wing twists and thicknesses) along with a realization of the random inputs define a wing for the aircraft that is trimmed for level flight and the augmented numerical model outputs the drag and stresses for this wing at level flight. This illustrates that a numerical model for deterministic optimization may not always make sense in the presence of uncertainties.

We further modify the problem by removing the rib and skin thicknesses from the

[3] The aspect ratio is a natural choice for the design variables as it provides a trade-off between aerodynamics and structural loads, but it is not included in the current free-form deformation parameterization of the wing.

Table 4.4: Triangular distributions of the 7 random inputs for the Bombardier Q400 wing.

Random Variable	Lower Limit	Mode	Upper Limit
Maximum take-off weight [kg]	29257 × 0.9	29257	29257 × 1.1
Fraction of weight not including wing	0.83	0.86	0.89
Cruise Mach number	0.55	0.6	0.65
Material density [kg/m^3]	2810 × 0.95	2810	2810 × 1.05
Material elastic modulus [GPa]	70 × 0.9	70	70 × 1.1
Material poisson ratio	0.31	0.33	0.35
Material yield stress [MPa]	370 × 0.9	370	370 × 1.1

design variables and their corresponding change-in-thickness constraints in order to reduce the size of the problem. Therefore, the robust optimization problem has 46 design variables (8 twist angles, 19 forward spar thicknesses, 19 aft spar thicknesses) and 40 constraints (4 stress constraints and 36 change in thickness constraints). Let the vector of design variables be $\mathbf{x}$ with lower and upper bounds $\mathbf{x}_L$ and $\mathbf{x}_U$, respectively, and let $\mathbf{u}$ be a realization of the random inputs $\mathbf{U}(\omega)$ listed in Table 4.4. The four stress constraints are denoted as

$$\begin{aligned}
\mathrm{ks}_1(\mathbf{x},\mathbf{u}) &= \text{KS function for rib stress} - \text{yield stress} \leq 0 \\
\mathrm{ks}_2(\mathbf{x},\mathbf{u}) &= \text{KS function for spar stress} - \text{yield stress} \leq 0 \\
\mathrm{ks}_3(\mathbf{x},\mathbf{u}) &= \text{KS function for upper surface skin stress} - \text{yield stress} \leq 0 \\
\mathrm{ks}_4(\mathbf{x},\mathbf{u}) &= \text{KS function for lower surface skin stress} - \text{yield stress} \leq 0.
\end{aligned}$$

The change-in-thickness constraints are linear and are represented as the matrix $\mathbf{K}$. In the presence of random inputs, the objective of the robust optimization problem is formulated as the mean + λ standard deviations of the drag. The stress constraints are formulated as mean + λ standard deviations ≤ 0. We chose $\lambda = 2$ for both the objective and the constraints. The change-in-thickness constraints are deterministic and do not need to be modified. Therefore, the robust wing optimization problem is

$$
\begin{aligned}
\min_{\mathbf{x}_L \le \mathbf{x} \le \mathbf{x}_U} \quad & \mathbb{E}\left[\mathrm{drag}(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{drag}(\mathbf{x}, \mathbf{U}(\omega))\right]} \\
\text{s.t.} \quad & \mathbb{E}\left[\mathrm{ks}_1(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{ks}_1(\mathbf{x}, \mathbf{U}(\omega))\right]} \le 0 \\
& \mathbb{E}\left[\mathrm{ks}_2(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{ks}_2(\mathbf{x}, \mathbf{U}(\omega))\right]} \le 0 \\
& \mathbb{E}\left[\mathrm{ks}_3(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{ks}_3(\mathbf{x}, \mathbf{U}(\omega))\right]} \le 0 \\
& \mathbb{E}\left[\mathrm{ks}_4(\mathbf{x}, \mathbf{U}(\omega))\right] + \lambda\sqrt{\mathrm{Var}\left[\mathrm{ks}_4(\mathbf{x}, \mathbf{U}(\omega))\right]} \le 0 \\
& \mathbf{K}\mathbf{x} \le 0.
\end{aligned}
$$

4.2.2 Numerical Results

We employ the constrained optimization by linear approximation algorithm (COBYLA) [41] developed for constrained optimization problems without analytical derivatives. Three levels of discretization of the wing are available to the aerostructural solver as shown in Table 4.5, making possible a multifidelity approach to solve the robust optimization problem. We select the aerostructural solver applied to the medium discretization as the high-fidelity model and the aerostructural solver applied to the coarse discretization as the low-fidelity model.[4] The ratio of average computational cost between the high-fidelity model and the low-fidelity model, w, is approximately 4. The optimization is conducted with the objective and constraint functions evaluated using the combined estimator and with the objective and constraint functions evaluated using the information reuse estimator. We do not have the regular Monte Carlo estimator as a comparison due to unaffordable computational cost. In both cases, the tolerance on the RMSE of the objective and constraint function estimators is fixed at 2×10^{-4}.

The convergence of the objective as a function of the cumulative computational effort is shown in Figure 4-8. The computational effort is the number of high-fidelity

[4] As future work, we may consider utilizing all three levels of discretization by extending the multifidelity estimator to two or more low-fidelity models. However, it is important to investigate if the additional low-fidelity models contribute enough unique approximate information about the high-fidelity model to justify the extra complexity and cost.

Table 4.5: Number of degrees of freedom for different discretizations of the aerodynamic and structural elements. The approximate evaluation time for the aerostructural solver wrapped with the secant method is based on a PC with 16 processors at 2.93 GHz each.

		Coarse	Medium	Fine
Aerodynamic panels	Chordwise	30	34	70
	Spanwise	20	40	80
	Wake	20	40	60
	Total panels	1000	2960	10,400
Structural elements	Chordwise	5	8	16
	Spanwise	30	80	320
	Thickness	4	6	12
	Total d.o.f.	5624	14,288	57,152
Approximate evaluation time [s]		6	24	350

model evaluations for the information reuse estimator and the equivalent number of high-fidelity model evaluations for the combined estimator. The initial wing drag coefficient has a mean of 0.1254 and a standard deviation of 0.008974 and the final wing drag coefficient has a mean of 0.1239 and a standard deviation of 0.008515—a reduction of about 1.2% in expected wing drag and a reduction of 5.1% in the standard deviation. In Figure 4-9, we plot the computational effort required to compute the combined estimator and the information reuse estimator at each optimization iteration. As discussed in previous examples, the information reuse estimator requires relatively large amount of computational effort during the first few optimization iterations. It can be seen that the main benefit of the combined estimator here is to reduce the computational effort during these first few optimization iterations by leveraging the cheaper low-fidelity model. The solution was obtained in 13.4 days using the information reuse estimator and in 9.7 days using the combined estimator on a PC with 16 processors at 2.93 GHz each. If the 90% computational savings of the combined estimator over the regular Monte Carlo estimator seen in the acoustic horn robust optimization example in §3.4.2 is extended to this problem, the solution would have taken about 3.2 months to obtain using the regular Monte Carlo estimator.

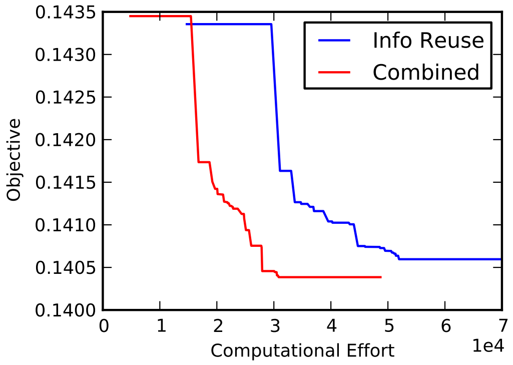
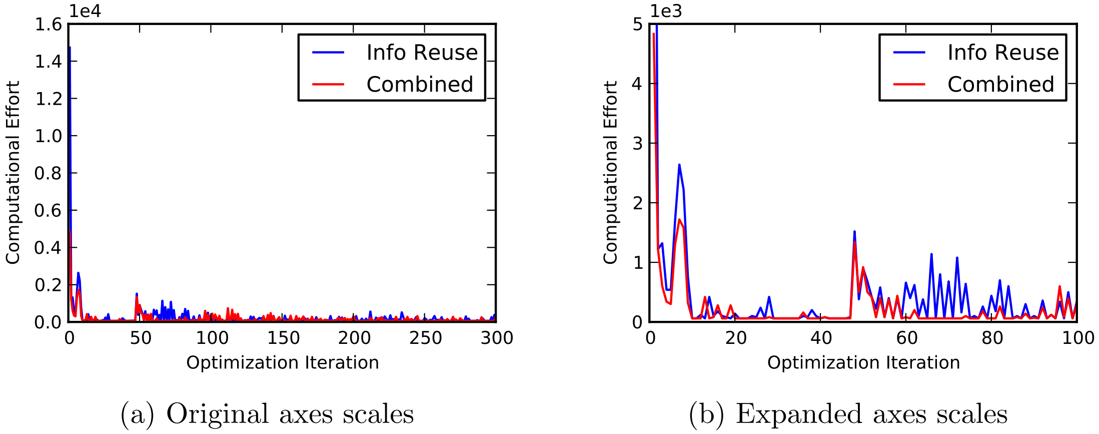


Figure 4-8: Comparison of convergence histories for the Bombardier Q400 wing optimization using the information reuse estimator and the combined estimator. The objective is mean + λ standard deviations of the drag coefficient. The optimization algorithm is COBYLA.



(a) Original axes scales

(b) Expanded axes scales

Figure 4-9: Computational effort per optimization iteration versus optimization iteration for the Bombardier Q400 wing optimization. The scales of the axes has been expanded in (b) to show the comparison more clearly.

4.3 Chapter Summary

In this chapter, we presented two application problems that make use of the estimators developed in Chapter 2 and Chapter 3. The conceptual aircraft design problem, due to the inherent uncertainties in projecting technological improvements, is a prime example to illustrate the importance of accounting for uncertainties in the conceptual design process. We demonstrated 90% reduction in computational cost using the information reuse estimator over the regular Monte Carlo estimator in the robust optimization problem and showed how the information reuse estimator can be efficiently adapted to solve multiple robust optimization problems for risk-performance trade studies.

In the wing design problem, we showed that numerical models requiring a discretization of an underlying geometry readily provide models of different levels of fidelity that can be utilized efficiently by multifidelity methods. Depending on the amount of control available over the discretization, it is also possible to employ the multilevel Monte Carlo method to achieve a faster convergence rate than the estimators presented here [15]. We demonstrated that the combined estimator, incorporating features from both the multifidelity estimator and the information reuse estimator, allows a large-scale robust optimization problem to be solved in a reasonable turnaround time of 9.7 days on a PC with 16 processors at 2.93 GHz each. Further development of this problem includes extending the free-form deformation parameterization of the wing to cover other geometric changes and incorporating distributed uncertainties in the structural properties along the wing. Distributed uncertainties is a more realistic scenario than uncertainties in the bulk material properties and may also be used to represent degradation in the wing structure. Since the estimators developed in this thesis are based on Monte Carlo sampling, they are well-suited to handle this high-dimensional (in the stochastic space) problem.

Chapter 5

Conclusions

In this thesis, we addressed the need for a general optimization under uncertainty approach that leverages inexpensive surrogate models and approximate information to reduce computational cost. To conclude, we summarize the developments and highlight the main contributions of this work. We close with suggestions for continuing work on this topic.

5.1 Thesis Summary

In many engineering applications, a suite of models is available to predict the outputs of interest. In Chapter 2, we utilized these multifidelity models to estimate the statistics of the high-fidelity model at reduced computational cost. There is no restriction on the types of multifidelity models—from different levels of physics to data-fit surrogates—but the multifidelity estimator is more efficient than the regular Monte Carlo estimator only when the low-fidelity model is cheaper to evaluate than the high-fidelity model and the outputs of the low-fidelity model are correlated with the outputs of the high-fidelity model. The low cost of the low-fidelity model allows us to calculate a more accurate estimate of the statistic of the low-fidelity model than is affordable with the high-fidelity model. The more accurate estimate of the statistic of the low-fidelity model is used to compute a correction term based on the control variate method to improve the accuracy of the estimate of the statistic of the high-fidelity model without requiring additional high-fidelity model evaluations. In the numerical results for the acoustic horn robust optimization example, we have demonstrated 85% reduction in computational cost when we make use of a reduced order model of the horn acoustics that is 40 times cheaper to evaluate than the high-fidelity finite element model.

To further reduce computational cost, we take advantage of the optimization under uncertainty setup in which the statistics of the high-fidelity model are estimated at many different vectors of design variables. In Chapter 3, we reused the freely available estimates of statistics from past optimization iterations (free in the sense that they are sunk cost) to improve the accuracy of the estimates of the statistics at the current optimization iteration without requiring additional model evaluations at the current optimization iteration. We showed that, for a model sufficiently smooth in the design space, the correlation between the model outputs at two vectors design variables increases as the distance between the two vectors of design variables decreases. This implies that as the optimization progresses towards the optimal solution and the steps taken in the design space become shorter and shorter, the information reuse estimator becomes more and more efficient. We also developed the combined estimator to take advantage of both the information reuse estimator and the multifidelity estimator to perform multifidelity optimization under uncertainty. In the acoustic horn robust optimization example, we have demonstrated 90% computational savings using the combined estimator.

In Chapter 4, we applied our approaches to two practical aerospace applications. In conceptual aircraft design, we demonstrated the need to consider uncertainties in the design process when making projections about the improvements in technologies. We showed that a simple adaptation of the information reuse estimator can be used to efficiently solve multiple robust optimization problems, making it computationally feasible to perform risk-performance trade studies. We also demonstrated that the information reuse estimator and the combined estimator enables a large-scale robust wing optimization problem to be solved in a reasonable turnaround time of 9.7 days on a 16-processor desktop machine, paving the way to accounting for degradation or damage to the wing that may be represented as distributed uncertainties in structural properties.

The methods we developed—the multifidelity estimator, the information reuse estimator, and the combined estimator—are not much more difficult to implement than the regular Monte Carlo estimator. They can make use of models that are provided as closed "black-boxes" and are embarrassingly parallelizable. Furthermore, they do not require a priori knowledge about the models; if the correlation turns out to be poor, the computational effort is not worse than that of regular Monte Carlo simulation. Although all of the estimators share the same slow convergence rate as the regular Monte Carlo estimator, the reduction in the root mean square error can be significant enough, as demonstrated in the numerical examples, to mitigate this disadvantage unless very high accuracy is needed. On the other hand, the methods are suitable for problems with a very large number of uncertain parameters. They represent a major contribution towards enabling rigorous accounting and mitigation of uncertainties in large-scale industrial applications.

5.2 Thesis Contributions

The main contributions of this work are:

- a new multifidelity approach to Monte Carlo simulation that has demonstrated reduction in computational cost of propagating uncertainties through an expensive high-fidelity model;

- a new method to estimate statistics during optimization under uncertainty that leverages the model autocorrelation in the design space to reuse information and reduce computational cost;

- non-intrusive, generally applicable, and easily implementable algorithms to enable practical risk-performance trade studies in engineering design problems with parameter uncertainties and variabilities in the numerical model; and

- a demonstration of practical design turnaround time for optimization under uncertainty in a large-scale high-fidelity engineering application.

5.3 Future Work

There are three potential thrusts of further research: in extending the capabilities and efficiencies of the statistic estimators, in developing optimization algorithms for noisy problems, and in advancing optimization under uncertainty in large-scale high-fidelity engineering problems. An apparent extension to the multifidelity estimator is to consider two or more low-fidelity models. While it is simple to modify the control variate formulation to include several auxiliary random variables, it is not immediately clear that the additional low-fidelity models will always provide sufficient additional benefit. This is because the multiple low-fidelity models are likely to be correlated with each other, and thus provide less unique new information for the high-fidelity model. Analogous considerations can be made for the information reuse estimator to reuse data from two or more past optimization iterations. The estimators may also be combined with other variance reduction techniques, including the StackedMC method [50]. Finally, it may be insightful to investigate whether the control variate based approaches can be generalized to other statistical methods beyond Monte Carlo simulation, particularly those that exhibit faster convergence rate for smooth problems. The challenge is in determining which aspect of the high-fidelity model output should be correlated with the surrogate (i.e., the low-fidelity model output or the high-fidelity model output at a different design point). For example, for the control variate method to be effective in quasi-Monte Carlo simulation using low-discrepancy sequences, the high frequency Fourier components or some derivatives of the high-fidelity model output should be correlated with those of the surrogate [21]. A similar situation is encountered in the case of polynomial chaos expansion [37].

While we focused the work of this thesis on developing efficient statistic estimators to be used in optimization under uncertainty, it is also important to advance the state of the art of optimization algorithms for problems with noisy objective and constraints. One path is to investigate stochastic approximation algorithms with general constraints using, for example, penalty functions [51], but this must be extended to include noisy evaluations of the constraint functions. Another avenue of research is to increase the robustness of derivative-free optimization algorithms to noise. One approach is to make the dynamic accuracy framework [6] less conservative and more efficient by tailoring it to the case of random noise—for example, accepting or rejecting noise probabilistically based on the comparison of the random noisy function values.

The final thrust is to deal with practical issues that, in additional to computational cost, hold back systematic considerations of uncertainties in engineering design. As demonstrated in the aircraft conceptual design problem, a numerical model that is set up for deterministic optimization may not be appropriate for optimization under uncertainty. It may be necessary to rethink the numerical model so that each realization of the random inputs define a physical system that makes sense. Finally the numerical models should be developed such that the properties of the system, including distributed properties, are easily parameterized. While these issues are based on the experiences in the application problems described in this thesis, more work must be done to address similar issues in problem formulation and setup for large-scale high-fidelity engineering design problems.

Appendix A

Autocorrelation Derivation

In this appendix, we derive the result

$$\begin{aligned}&\operatorname{Corr}\left[M(x+\Delta x,\mathbf{U}(\omega)),M(x,\mathbf{U}(\omega)))\right]\\&\approx 1-\frac{1-\operatorname{Corr}\left[M'(x,\mathbf{U}(\omega)),M(x,\mathbf{U}(\omega))\right]^2}{2\operatorname{Var}\left[M(x,\mathbf{U}(\omega))\right]/\operatorname{Var}\left[M'(x,\mathbf{U}(\omega))\right]}\Delta x^2\end{aligned}$$

for $|\Delta x| \ll 1$ presented in §3.1.1. To simplify notation, we drop $\mathbf{U}(\omega)$ from the argument of the model M.

Let the model be twice differentiable in x for all realizations of $\mathbf{U}(\omega)$. For some small Δx, we apply the Taylor expansion in x to obtain

$$M(x+\Delta x) \approx M(x) + M'(x)\Delta x + M''(x)\frac{\Delta x^2}{2},$$

where $M'(x) = \frac{\partial M(x)}{\partial x}$ and $M''(x) = \frac{\partial^2 M(x)}{\partial x^2}$. Taking the expectation of both sides, we get

$$\mu_M(x+\Delta x) \approx \mu_M(x) + \mu_{M'}(x)\Delta x + \mu_{M''}(x)\frac{\Delta x^2}{2},$$

where $\mu_M(x) = \mathbb{E}\left[M(x)\right]$, $\mu_{M'} = \mathbb{E}\left[M'(x)\right]$, and $\mu_{M''} = \mathbb{E}\left[M''(x)\right]$.

We first derive the covariance between $M(x+\Delta x)$ and $M(x)$:

$$\begin{aligned}
&\mathrm{Cov}\left[M(x+\Delta x), M(x)\right] \\
=\ &\mathbb{E}\left[\{M(x+\Delta x)-\mu_M(x+\Delta x)\}\{M(x)-\mu_M(x)\}\right] \\
\approx\ &\mathbb{E}\left[\{M(x)-\mu_M(x)\}^2\right]+\mathbb{E}\left[\{M'(x)-\mu_{M'}(x)\}\{M(x)-\mu_M(x)\}\right]\Delta x \\
&+\mathbb{E}\left[\{M''(x)-\mu_{M''}(x)\}\{M(x)-\mu_M(x)\}\right]\frac{\Delta x^2}{2} \\
=\ &\mathrm{Var}\left[M(x)\right]+\mathrm{Cov}\left[M'(x), M(x)\right]\Delta x+\mathrm{Cov}\left[M''(x), M(x)\right]\frac{\Delta x^2}{2} \\
=\ &\mathrm{Var}\left[M(x)\right]\left\{1+\frac{\mathrm{Cov}\left[M'(x), M(x)\right]}{\mathrm{Var}\left[M(x)\right]}\Delta x+\frac{\mathrm{Cov}\left[M''(x), M(x)\right]}{\mathrm{Var}\left[M(x)\right]}\frac{\Delta x^2}{2}\right\}.
\end{aligned}$$

Next, we derive the variance of $M(x+\Delta x)$ in a similar manner:

$$\begin{aligned}
&\mathrm{Var}\left[M(x+\Delta x)\right] \\
=\ &\mathbb{E}\left[\{M(x+\Delta x)-\mu_M(x+\Delta x)\}^2\right] \\
\approx\ &\mathrm{Var}\left[M(x)\right]+\mathrm{Var}\left[M'(x)\right]\Delta x^2+\mathrm{Var}\left[M''(x)\right]\frac{\Delta x^4}{4}+2\,\mathrm{Cov}\left[M'(x), M(x)\right]\Delta x \\
&+2\,\mathrm{Cov}\left[M''(x), M(x)\right]\frac{\Delta x^2}{2}+2\,\mathrm{Cov}\left[M''(x), M'(x)\right]\frac{\Delta x^3}{2}.
\end{aligned}$$

Using the formula for the Taylor expansion of the inverse square root

$$\frac{1}{\sqrt{a+bz+cz^2+dz^3+ez^4}}\approx\frac{1}{a^{1/2}}-\frac{b}{2a^{3/2}}z+\frac{3b^2-4ac}{8a^{5/2}}z^2,$$

we obtain

$$\begin{aligned}
&\frac{1}{\sqrt{\mathrm{Var}\left[M(x+\Delta x)\,\mathrm{Var}\left[M(x)\right]\right]}} \\
\approx\ &\frac{1}{\mathrm{Var}\left[M(x)\right]}-\frac{\mathrm{Cov}\left[M'(x), M(x)\right]}{\mathrm{Var}\left[M(x)\right]^2}\Delta x \\
&+\frac{3\,\mathrm{Cov}\left[M'(x), M(x)\right]^2-\mathrm{Var}\left[M(x)\right]\{\mathrm{Var}\left[M'(x)\right]+\mathrm{Cov}\left[M''(x), M(x)\right]\}}{2\,\mathrm{Var}\left[M(x)\right]^3}\Delta x^2
\end{aligned}$$

Therefore, the correlation coefficient between $M(x+\Delta x)$ and $M(x)$, omitting terms of order Δx^3 or higher, is

$$\begin{aligned}
&\operatorname{Corr}\left[M(x+\Delta x), M(x)\right] \\
&= \frac{\operatorname{Cov}\left[M(x+\Delta x), M(x)\right]}{\sqrt{\operatorname{Var}\left[M(x+\Delta x)\operatorname{Var}\left[M(x)\right]\right]}} \\
&\approx 1 - \frac{\operatorname{Cov}\left[M'(x), M(x)\right]}{\operatorname{Var}\left[M(x)\right]}\Delta x \\
&\quad + \frac{3\operatorname{Cov}\left[M'(x), M(x)\right] - \operatorname{Var}\left[M(x)\right]\{\operatorname{Var}\left[M'(x)\right] + \operatorname{Cov}\left[M''(x), M(x)\right]\}}{2\operatorname{Var}\left[M(x)\right]^2}\Delta x^2 \\
&\quad + \frac{\operatorname{Cov}\left[M'(x), M(x)\right]}{\operatorname{Var}\left[M(x)\right]}\Delta x - \frac{\operatorname{Cov}\left[M'(x), M(x)\right]^2}{\operatorname{Var}\left[M(x)\right]^2}\Delta x^2 + \frac{\operatorname{Cov}\left[M''(x), M(x)\right]}{2\operatorname{Var}\left[M(x)\right]}\Delta x^2 \\
&= 1 - \frac{\operatorname{Var}\left[M'(x)\right]\operatorname{Var}\left[M(x)\right] - \operatorname{Cov}\left[M'(x), M(x)\right]^2}{2\operatorname{Var}\left[M(x)\right]^2}\Delta x^2 \\
&= 1 - \frac{1 - \operatorname{Corr}\left[M'(x), M(x)\right]^2}{2\operatorname{Var}\left[M(x)\right]/\operatorname{Var}\left[M'(x)\right]}\Delta x^2.
\end{aligned}$$

Bibliography

[1] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Approximation and Model Management in Aerodynamic Optimization with Variable Fidelity Models. *AIAA Journal of Aircraft*, 38(6):1093–1101, 2001.

[2] C. Audet and J. E. Dennis Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[3] H. G. Beyer and B. Sendhoff. Robust Optimization—A Comprehensive Survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.

[4] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A Rigorous Framework for Optimization of Expensive Functions by Surrogates. *Structural and Multidisciplinary Optimization*, 17(1):1–13, 1999.

[5] T. Borogovac and P. Vakili. Control Variate Technique: A Constructive Approach. In *Winter Simulation Conference*, pages 320–327, Miami, FL, December 2008.

[6] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, PA, 2000.

[7] A. R. Conn, K. Scheinberg, and P. L. Toint. A Derivative Free Optimization Algorithm in Practice. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-1998-4718, St. Louis, MO, September 1998.

[8] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. SIAM, Philadelphia, PA, 2009.

[9] A. P. Dempster. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[10] M. Drela. Design Derivers of Energy-Efficient Transport Aircraft. *SAE International Journal of Aerospace*, 4(2):602–618, 2011.

[11] C. Dribusch, S. Missoum, and P. Beran. A Multifidelity Approach for the Construction of Explicit Decision Boundaries: Application to Aeroelasticity. *Structural and Multidisciplinary Optimization*, 42(5):693–705, 2010.

[12] D. Dubois and H. Prade. Possibility Theory, Probability Theory and Multiple-Valued Logics: A clarification. *Annals of Mathematics and Artificial Intelligence*, 32(1):35–66, 2001.

[13] J. L. Eftang, D. B. P. Huynh, D. J. Knezevic, and A. T. Patera. A Two-Step Certified Reduced Basis Method. *SIAM Journal of Scientific Computing*, 51(1):28–58, 2012.

[14] M. Emsermann and B. Simon. Improving Simulation Efficiency with Quasi Control Variates. *Stochastic Models*, 18(3):425–448, 2002.

[15] M. B. Giles. Multilevel Monte Carlo Path Simulation. *Operations Research*, 56(3):607–617, 2008.

[16] E. M. Greitzer, P. A. Bonnefoy, E. de la Rosa Blanco, C. S. Dorbian, M. Drela, D. K. Hall, R. J. Hansman, J. I. Hileman, R. H. Liebeck, J. Lovegren, P. Mody, J. A. Pertuze, S. Sato, Z. S. Spakovszky, C. S. Tan, J. S. Hollman, J. E. Duda, N. Fitzgerald, J. Houghton, J. L. Kerrebrock, G. F. Kiwada, D. Kordonowy, J. C. Parrish, J. Tylko, E. A. Wen, and W. K. Lord. N+3 Aircraft Concept Designs and Trade Studies, Volume 1. Final Report NASA/CR-2010-216794/VOL1, NASA Glenn Research Center, Cleveland, OH, 2010.

[17] E. M. Greitzer, P. A. Bonnefoy, E. de la Rosa Blanco, C. S. Dorbian, M. Drela, D. K. Hall, R. J. Hansman, J. I. Hileman, R. H. Liebeck, J. Lovegren, P. Mody, J. A. Pertuze, S. Sato, Z. S. Spakovszky, C. S. Tan, J. S. Hollman, J. E. Duda, N. Fitzgerald, J. Houghton, J. L. Kerrebrock, G. F. Kiwada, D. Kordonowy, J. C. Parrish, J. Tylko, E. A. Wen, and W. K. Lord. N+3 Aircraft Concept Designs and Trade Studies, Volume 2, Design Methodologies for Aerodynamics, Structures, Weight, and Thermodynamic Cycles. Final Report NASA/CR-2010-216794/VOL2, NASA Glenn Research Center, Cleveland, OH, 2010.

[18] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods.* Methuen, London, UK, 1964.

[19] S. Heinrich. Multilevel Monte Carlo Methods. In S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, pages 58–67. Springer-Verlag, Berlin, Germany, 2001.

[20] T. C. Hesterberg and B. L. Nelson. Control Variates for Probability and Quantile Estimation. *Management Science*, 44(9):1295–1312, 1998.

[21] F. J. Hickernell, C. Lemieux, and A. B. Owen. Control Variates for Quasi-Monte Carlo. *Statistical Science*, 20(1):1–31, 2005.

[22] K. C. Kapur and L. R. Lamberson. *Reliability in Engineering Design.* John Wiley & Sons, New York, NY, 1977.

[23] C. T. Kelley. *Implicit Filtering.* SIAM, Philadelphia, PA, 2011.

[24] G. J. Kennedy and J. R. R. A. Martins. Parallel Solution Methods for Aerostructural Analysis and Design Optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2010-9308, Fort Worth, TX, September 2010.

[25] M. C. Kennedy and A. O'Hagan. Predicting the Output from a Complex Computer Code When Fast Approximations Are Available. *Biometrika*, 87(1):1–13, 2000.

[26] M. C. Kennedy and A. O'Hagan. Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[27] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. A CAD-Free Approach to High-Fidelity Aerostructural Optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2010-9231, Fort Worth, TX, September 2010.

[28] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, Reading, MA, 3 edition, 1998.

[29] N. Kuschel and R. Rackwitz. Two Basic Problems in Reliability-Based Structural Optimization. *Mathematical Methods of Operations Research*, 46(3):309–333, 1997.

[30] J. Li and D. Xiu. Evaluation of Failure Probability via Surrogate Models. *Journal of Computational Physics*, 229(23):8966–8980, 2010.

[31] W.-K. Mak, D. P. Morton, and R. K. Wood. Monte Carlo Bounding Techniques for Determining Solution Quality in Stochastic Programs. *Operations Research Letters*, 24(1):47–56, 1999.

[32] A. March and K. Willcox. Constrained Multifidelity Optimization Using Model Calibration. *Structural and Multidisciplinary Optimization*, 46(1):93–109, 2012.

[33] A. March and K. Willcox. Provably Convergent Multifidelity Optimization Algorithm not Requiring High-Fidelity Derivatives. *AIAA Journal*, 50(5):1079–1089, 2012.

[34] A. I. March. *Multifidelity Methods for Multidisciplinary System Design*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2012.

[35] B. L. Nelson. On Control Variate Estimators. *Computers & Operations Research*, 14(3):219–225, 1987.

[36] B. L. Nelson, B. W. Schmeiser, M. R. Taaffe, and J. Wang. Approximation-Assisted Point Estimation. *Operations Research Letters*, 20(3):109–118, 1997.

[37] L. W. T. Ng and M. S. Eldred. Multifidelity Uncertainty Quantification Using Nonintrusive Polynomial Chaos and Stochastic Collocation. In *14th AIAA Non-Deterministic Approaches Conference*, number AIAA 2012-1852, Honolulu, HI, April 2012.

[38] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin. Error and Uncertainty in Modeling and Simulation. *Reliability Engineering & System Safety*, 75(3):333–357, 2002.

[39] G. J. Park, T. H. Lee, K. H. Lee, and K. H. Hwang. Robust Design: An Overview. *AIAA Journal*, 44(1):181–191, 2006.

[40] R. Pasupathy, B. W. Schmeiser, M. R. Taaffe, and J. Wang. Control-Variate Estimation Using Estimated Control Means. *IIE Transactions*, 44(5):381–385, 2012.

[41] M. J. D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. *Advances in Optimization and Numerical Analysis*, 7:51–67, 1994.

[42] M. J. D. Powell. The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives. Technical Report 2009/NA06, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 2009.

[43] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced Basis Approximation and A Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[44] B. W. Schmeiser, M. R. Taaffe, and J. Wang. Biased Control-Variate Estimation. *IIE Transactions*, 33(3):219–228, 2001.

[45] G. Shafer. *A Mathematical Theory of Evidence.* Princeton university press Princeton, Princeton, NJ, 1976.

[46] J. C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[47] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* John Wiley & Sons, Hoboken, NJ, 2003.

[48] A. Speight. A Multilevel Approach to Control Variates. *Journal of Computational Finance*, 12(4):3–27, 2009.

[49] V. Torczon. On the Convergence of Pattern Search Algorithms. *SIAM Journal on optimization*, 7(1):1–25, 1997.

[50] B. Tracey, D. Wolpert, and J. J. Alonso. Using Supervised Learning to Improve Monte Carlo Integral Estimation. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Material Conference*, number AIAA 2011-1843, Denver, CO, April 2011.

[51] I.-J. Wang and J. C. Spall. A Constrained Simultaneous Perturbation Stochastic Approximation Algorithm Based on Penalty Functions. In *American Control Conference*, San Diego, CA, June 1999.

[52] U. Wolff. Monte Carlo Errors with Less Errors. *Computer Physics Communications*, 156(2):143–153, 2004.

[53] L. A. Zadeh. Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.