

A BAYESIAN APPROACH TO COMPUTER MODEL CALIBRATION AND
MODEL-ASSISTED DESIGN

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

by
Carl Ehrett
December 2020

Accepted by:
Dr. Andrew Brown, Committee Chair
Dr. Sez Atamturktur
Dr. Chris Kitchens
Dr. Chris McMahan

ABSTRACT

Computer models of phenomena that are difficult or impossible to study directly are critical for enabling research and assisting design in many areas. In order to be effective, computer models must be calibrated so that they accurately represent the modeled phenomena. There exists a rich variety of methods for computer model calibration that have been developed in recent decades. Among the desiderata of such methods is a means of quantifying remaining uncertainty after calibration regarding both the values of the calibrated model inputs and the model outputs. Bayesian approaches to calibration have met this need in recent decades. However, limitations remain. Whereas in model calibration one finds point estimates or distributions of *calibration inputs* in order to induce the model to reflect reality accurately, interest in a computer model often centers primarily on its use for model-assisted design, in which the goal is to find values for *design inputs* to induce the modeled system to approximate some target outcome. Existing Bayesian approaches are limited to the first of these two tasks. The present work develops an approach adapting Bayesian methods for model calibration for application in model-assisted design. The approach retains the benefits of Bayesian calibration in accounting for and quantifying all sources of uncertainty. It is capable of generating a comprehensive assessment of the Pareto optimal inputs for a multi-objective optimization problem. The present work shows that this approach can apply as a method for model-assisted design using a previously calibrated system, and can also serve as a method for model-assisted design using a model that still requires calibration, accomplishing both ends simultaneously.

DEDICATION

For my father, who made me see that this was possible, and for my wife, without whom it would not have been.

ACKNOWLEDGMENTS

Many thanks to Dr. Andrew Brown for guiding me toward and through the writing of this dissertation.

I also gratefully acknowledge grant CMMI-1934438 from the National Science Foundation (NSF). I was supported by fellowships through Department of Education GAANN grant P200A150310 and NSF NRT grant 1633608. During the writing of this work Dr. Brown was also supported by NSF grants EEC-1744497 and OIA-1826715.

TABLE OF CONTENTS

	Page
TITLE PAGE	i
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGMENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER	
I. INTRODUCTION	1
Computer model calibration	1
Calibration versus model-assisted design	9
Model-assisted design	10
Gaussian process metamodels	11
Optimization	13
Applications	16
References	18
II. MULTI-OBJECTIVE ENGINEERING DESIGN VIA COMPUTER MODEL CALIBRATION	29
Introduction	32
Counterfactual Bayes	37
Calibration for design	38
Simulated example	47
Wind turbine material design application	49
Discussion	59
References	63
Appendix: Turbine blade finite element model	72
Appendix: Surrogate model validation	73
III. A UNIFIED FRAMEWORK FOR COMPUTER MODEL CALIBRATION AND ENGINEERING DESIGN	75

Table of Contents (Continued)

	Page
Introduction.....	78
Dual calibration to target outcomes	85
Dependence of θ_c on θ_d	92
Case study application: vibration isolation design.....	103
Conclusion	116
References.....	118
Appendix: Validation of DCTO.....	124
IV. CONCLUSION.....	128
Benefits	128
Summary of chapter two.....	129
Summary of chapter three.....	132
Recommendations.....	134
Future directions	136
References.....	137

LIST OF TABLES

Table	Page
2.1 Turbine blade surrogate hyperparameter estimates	52
3.1 RMSEs for simulation calibration and design	101
3.2 Values of components in vibration isolation test rig	107
3.3 Vibration isolation experiment tests and results	109
3.4 Vibration isolation simulation inputs and results.....	113

LIST OF FIGURES

Figure	Page
2.1 Two choices of target outcomes for CTO.....	47
2.2 True two-dimensional profile outputs of the five-dimensional simulated example model.....	48
2.3 Posterior draws from CTO in the simulated example.....	50
2.4 Estimated Pareto front with target outcome.....	54
2.5 Posterior distribution of θ	55
2.6 Approximate prior and posterior marginal predictive densities for each of the three turbine outputs	55
2.7 Estimated Pareto front of the turbine system with Quantified uncertainties and NSGA-II results.....	57
2.8 Estimated Pareto front for multiple turbine systems in a variety of design spaces with NSGA-II results.....	47
2.9 CX-100 blade model.....	72
2.10 Results of 10-fold cross validation of the GP emulator used for the turbine application.....	74
3.1 Example simulation model output	95
3.2 True value of the calibration parameter θ_c for each value in the domain of t_d	95
3.3 Simulation model output for all values of t_c and t_d	96
3.4 Simulation model outputs shown with various model discrepancies	98
3.5 Noisy observations of the simulation model.....	99
3.6 Design input values for observations under the adaptive sampling approach and space-filling design.....	100

List of Figures (Continued)

Figure	Page
3.7 Prior and posterior distributions of θ_c and θ_d with true/ optimal values for various simulation discrepancies	102
3.8 Physical test rig for vibration isolation design experiment.....	104
3.9 Schematic diagram of the vibration isolation test rig	105
3.10 Schematic diagram of the applied impulse force in the time domain	110
3.11 Five-times averaged acceleration response of the rigid mass in the time domain	110
3.12 The dynamic vibration system.....	112
3.13 The posterior distributions of the calibration and design inputs along with their priors	115
3.14 The posterior distributions of the model output at three levels of the operational domain, with priors.....	117
3.15 Prior and posterior distributions of θ_c and θ_d for validation of DCTO	125

Introduction

Carl Ehrett
Watt Family Innovation Center,
Clemson University

1 Computer model calibration

1.1 Computer experiments

Suppose that one wishes to improve one's understanding of, say, the movement of people in a crowd escaping from a building in a crisis situation. This is an example of an area in which field data, i.e. real-world observations, are extremely difficult to acquire. Merely assembling a crowd of research subjects in one place is costly and difficult. Asking them to flee a building may result in behaviors which are unlike those in real crisis situations – but which may nonetheless present unacceptable physical risk to the subjects. Inducing them to flee through the generation of a (real or apparent) crisis is similarly infeasible. Observational data are likewise scarce here, since panic-inducing crises are by their nature difficult to predict and chaotic in ways that hinder the orderly collection of data.

Such problems can also hinder much more modest research aims. The cost of a task as simple as measuring, e.g., the time required for balls of a variety of sizes to fall from a variety of heights, can be prohibitive depending upon one's available resources.

In the face of these difficulties, computer models offer a third alternative to the choice between attempting field data collection and giving up on the hope of progress. Using existing theory concerning human psychology and movement, it is possible to construct a computer model simulating the behavior of people evacuating from a large building (Thompson and Marchant, 1995), to allow one to observe simulated evacuation behaviors. With such a model, a user might input floor plans for each floor of a building, the locations of staircases and exits, the locations of occupants, the dimensions of their bodies, routes to exits, and so on. The user could then observe how long it takes each simulated occupant to exit the building. Similarly, a simple physics-based model can simulate the height-time curve of balls falling through the air (Gattiker et al., 2017), with the user specifying the height of the fall and the radius of the ball. Thus, computer models provide a means to collect data which might otherwise be inaccessible.

The study of computer models from a statistical perspective calls for specialized tools and techniques. Gaussian processes (GPs) are popular for modeling the output of computer code when the code is too computationally expensive to allow for repeated evaluation. The GP then provides a computationally inexpensive emulator for the computer code. There are three reasons the popularity of GPs as emulators: (1) The use of a GP does not require detailed foreknowledge of the approximate parametric form of the computer model. Researchers often lack such foreknowledge in the case of complex computer models. (2) GPs easily interpolate the observed data. This is an

advantage when the observations come from deterministic computer code that is free of observation error. (3) The variance of a GP provides a natural form of uncertainty quantification.

1.2 Computer model calibration

Again considering the hypothetical model of evacuation behavior, suppose a user wishes to compare two different proposed building codes to be enforced in a given area. One might use average interpersonal distance in a crisis evacuation as an input parameter for such a model, both to settle the initial physical distribution of people throughout the building and to influence their behavior during evacuation. It is well-established that customs of interpersonal distance vary across locales (Sorokowska et al., 2017). These values may be unknown for the case of a particular region, particularly in a crisis setting. Thus we may wish to find the true values for interpersonal distance in that region. With respect to the physics-based simulator, the model may require the user to estimate the drag coefficient of the ball. In both cases, the user will need to *calibrate* these parameters in the model.

Broadly, in model calibration, we may consider a model to be of the form $\eta(\mathbf{x}, \boldsymbol{\theta})$, where $(\mathbf{x}, \boldsymbol{\theta})$ comprise all inputs to the model. Input vector \mathbf{x} is the collection of inputs that are known and/or under the control of the researcher (in the evacuation example, this might include the building layout; in the freefall example, it might include the height of the fall). The vector of calibration inputs $\boldsymbol{\theta}$ is the collection of

parameters the values of which are unknown. These must be estimated for successful simulation. Thus where f describes the true system and \mathbf{y} our observations of that system, consider the model to be

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon(\mathbf{x}) \quad (1)$$

where $\delta(\cdot)$ describes the model discrepancy – i.e., the bias of the model as an estimate of the real system – and $\epsilon(\cdot)$ is a mean-zero observation error, often i.i.d. Gaussian. To undertake model calibration, one must have access to at least some observations of the real system, using them to “tune” the values of the calibration parameters.

Though the above examples relate to cases of estimating the true value of some parameter, calibration need not take this form. Calibration can also take the form of setting values which optimize the model’s faithfulness to reality, even when the values being set do not correspond to any particular real features. For example, a classification model may return scores which constitute some measure of confidence of category membership. A probability calibration of the model would involve mapping these confidence scores to values in the interval $[0, 1]$ such that the resulting transformed values are plausible estimates of the probability of category membership (Niculescu-Mizil and Caruana, 2005).

Given a set of data which may be used for calibration, the available calibration methods are diverse. Often, calibration is approached in an *ad hoc* manner, by manually searching for a set of values for the parameters of interest that cause the

model output to come acceptably close to the observed data. This manual approach is often improved by automating it as a grid search or stochastic search, where some measure of model performance is optimized in a brute force manner.

Even among more sophisticated approaches, calibration methods often yield only a point estimate of the appropriate value(s) for the calibration parameter(s). As a result, much interest in the past two decades has centered on Bayesian methods for model calibration. The appeal of a Bayesian approach to model calibration lies in the fact that the calibration parameters are a source of uncertainty for the model. This uncertainty should be quantified so that its effect on the model can be made explicit. One can thus use Bayesian methods to arrive at a posterior distribution of the calibration parameters which both balances our prior knowledge about the calibration parameters with what can be learned from the available data and also allows for accurate uncertainty quantification on the model outputs.

Trucano et al. (2006) provide a high-level overview of the relationship of model calibration to both model validation and sensitivity analysis. Much of the field of computer model calibration takes place either within the framework of, or as a response to, the work of Kennedy and O'Hagan (2001). This work established the dominant paradigm for Bayesian computer model calibration. In this dissertation we refer to that paradigm as the KOH framework. In KOH, one weds a set of experimental data to a computer model that stands in need of calibration, often mediated through a Gaussian process (GP) surrogate model when the original model is of high

computational complexity. Systematic discrepancy between the model and the true system can also be represented by a GP, and priors are placed on the calibration parameters. GP hyperparameters are either estimated via maximum likelihood estimation (MLE) or else are also assigned prior distributions. Via integration or some form of Markov-chain Monte Carlo (MCMC, Gelfand and Smith (1990)), a posterior distribution on the calibration parameters is obtained.

Of course, not all work in computer model calibration derives from the KOH framework. For example, Craig et al. (1997) provide an influential demonstration using Bayes linear methods to calibrate a model through a combination of expert judgments (about means and variances of model parameters) and information from runs of low-fidelity versions of the model. Contemporaneously to Kennedy and O'Hagan (2001), Cox et al. (2001) offer a similar calibration framework from a frequentist perspective, including the use of GPs as meta-models for computational efficiency. The authors use MLEs of calibration parameters and GP hyperparameters. Similarly, and more in response to the KOH framework, Loeppky et al. (2006) also provide an MLE-based alternative to KOH that is designed to improve the identifiability of the calibration parameters when model discrepancy is present (i.e., when the model is a systematically biased representation of the target phenomenon). More recently, Wong et al. (2014) describe general framework for a semi-parametric frequentist approach to model calibration, using bootstrapping to provide uncertainty quantification.

Far more common in the field of computer model calibration are extensions and

refinements of the KOH framework. Higdon et al. (2004) provide such an extension, adding uncertainty quantification to KOH calibration, to estimate remaining uncertainty regarding the values of the calibration parameter(s). Williams et al. (2006) further refine and exemplify this approach. Bayarri et al. (2007b) extend KOH to effect simultaneous validation and calibration of a computer model, while Bayarri et al. (2007a) apply this approach to functional data, using a hierarchical representation of coefficients of a wavelet basis. Paulo et al. (2012) further extend the work of Bayarri et al. (2007b) to the case of multivariate model output. While most work in this area assumes that the computer model is deterministic, Pratola and Chkrebtii (2018) provide an example of non-deterministic model calibration.

Paulo et al. (2012) are part of a diverse array of projects aiming to shore up a weakness of the KOH framework: its difficulty in accommodating high-dimensional data and/or large data sets in a computationally efficient way. Higdon et al. (2008) focus on meeting the computational challenges of high-dimensional, large data sets by using principal components basis representations for dimensionality reduction. Whereas that work is concerned with the dimensionality of model outputs, Drignei and Mourelatos (2012) are concerned rather with the model inputs. They increase the numerical stability of model calibration by using a global sensitivity analysis to reduce the model inputs' dimensionality. Bhat et al. (2010) demonstrate the application of the KOH framework to multivariate spatial output, and Pratola et al. (2013) apply KOH to nonstationary spatial-temporal field output. Higdon et al. (2013) adapt KOH

to employ an ensemble Kalman filter rather than GPs, for improved computational efficiency. Yuan and Ng (2013) offer a sequential approach to model calibration, in which each evaluation of the high-fidelity model is selected using the previous evaluations to minimize the resulting calibration uncertainty. Such an approach reduces the total number of required model evaluations and thereby the total computational cost of the calibration procedure.

Other works have endeavored to address another known weakness of KOH: poor identifiability of the calibration parameters in the face of model discrepancy. With respect to this problem, Brynjarsdóttir and O'Hagan (2014) emphasize the importance of strong priors on the model discrepancy term when performing model calibration. And for the purpose of improving identifiability over previous approaches, Gu and Wang (2018) propose a novel stochastic process that combines elements of GPs with L_2 calibration (in which the calibration parameter is chosen so as to minimize the L_2 norm of the discrepancy term).

Some works in the area of computer model calibration are premised upon broadening the conception of calibration beyond the idea of relating experimental data to a computer model of the experimental phenomenon. Model calibration can be seen more generally as a method for relating two or more different sources of data with varying costs and varying levels of fidelity. In this vein Kennedy and O'Hagan (2000) and Qian et al. (2006) explore methods for integrating “high” and “low” models to build a computationally efficient surrogate. Goh et al. (2013) extend KOH to cases

of more than two different levels of fidelity.

2 Calibration versus model-assisted design

“Design” used here means not experimental design, but rather refers to making engineering choices to try to achieve a desired outcome. Thus the sort of design considered here is related to the field of optimization. Where calibration involves setting input parameters to induce the model to approximate reality, design involves choosing input parameters to induce an engineered system to behave in some desired way. To understand the way the present work applies methods of computer model calibration, it is helpful to consider the relationship between the two tasks of calibration and design.

At the highest level, calibration can be conceived as follows. One has a model $\eta : \mathbb{R}^{p+q} \rightarrow \mathbb{R}^m$, as well as a set of data \mathbf{y} and corresponding p -dimensional inputs \mathbf{x}_y . We can thus consider η to be a function of two vectors, $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{t} \in \mathbb{R}^q$. Often (but not exclusively) \mathbf{y} is a set of observed experimental outcomes from a real phenomenon $f : \mathbb{R}^p \rightarrow \mathbb{R}^m$ where η is a model of f . One wishes to use \mathbf{y} to select values $\boldsymbol{\theta}$ such that setting $\mathbf{t} = \boldsymbol{\theta}$ induces the output of $\eta(\mathbf{x}_y, \boldsymbol{\theta})$ to approximate \mathbf{y} . The selection of these values is the calibration of η .

Similarly, model-assisted design can be conceived as follows. One has a model $\eta : \mathbb{R}^{p+q} \rightarrow \mathbb{R}^m$, as well as a set of data \mathbf{y} and corresponding p -dimensional inputs \mathbf{x}_y . We can thus consider η to be a function of two vectors, $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{t} \in \mathbb{R}^q$. Often (but not exclusively) \mathbf{y} is a set of target outcomes one wishes to achieve in

some system $f : \mathbb{R}^p \rightarrow \mathbb{R}^m$ where η is a model of f . One wishes to use \mathbf{y} to select values $\boldsymbol{\theta}$ such that setting $\mathbf{t} = \boldsymbol{\theta}$ induces the output of $\eta(\mathbf{x}_\mathbf{y}, \boldsymbol{\theta})$ to approximate \mathbf{y} . The selection of these values is model-assisted design using η .

Seen in this way, calibration and design are surprisingly similar undertakings. This raises the possibility of applying tools from one domain to problems in the other. One of the primary goals of the present work is to adapt the KOH framework for model-assisted design, and to demonstrate its utility and flexibility in that context.

3 Model-assisted design

The present work thus must be situated in the context of model-assisted design. In model-assisted design, one optimizes a model of the system of interest with respect to some objective function. The resulting optimal model inputs are treated as estimates of the optimal design settings for the system. The present work is intended to be applicable to problems of multi-objective design.

In the field of model- and metamodel-assisted design, Sacks et al. (1989) provide a very influential discussion of strategies to accommodate computer models of high computational complexity with GP surrogates. Following in this vein, Santner et al.'s (2003) foundational work serves as a focal point for much subsequent discussion of how best to learn from computer models in conjunction with physical experiments, for purposes including but not limited to engineering design. Currin et al. (1988) and Currin et al. (1991) develop a similar approach to that of Sacks et al 1989, but

with a Bayesian interpretation. Mitchell and Morris (1992) demonstrate the resulting methodology. Craig et al. (2001) discuss Bayesian methods for forecasting using a computer model under uncertainty. Simpson et al. (2008) provide an overview and retrospective of progress in the area of model- and metamodel-assisted design optimization since the work of Sacks et al. (1989), and highlight the value of employing multiple models with varying degrees of fidelity. Bartz-Beielstein and Zaefferer (2017) give a similar, more up-to-date discussion, focusing on discrete optimization problems. Westermann and Evins (2019) offer a comprehensive and illustrative discussion of the use of metamodels for optimal design of sustainable buildings, including a focus on sensitivity analysis and uncertainty quantification.

4 Gaussian process metamodels

The present work shares with many KOO-based, Bayesian optimization (BO)-based, and other approaches a reliance on GP metamodels as surrogates for computationally expensive models. Long popular in geostatistics (where GP regression is referred to as *kriging*; Cressie (2015)), the use of GPs as computer model surrogates was popularized by Sacks et al. (1989). GPs are attractive in this context due to their flexibility, and the ease with which they can interpolate observations of deterministic computer model output while providing closed-form expressions for uncertainty quantification. The application of GPs in this domain is further explored by Santner et al. (2003), who include discussion of the choice of correlation function to suit the

desired smoothness properties, and also include a discussion of hierarchical Gaussian random field models for cases when the user is not prepared to specify the desired smoothness. Diagnostic methods for validating a GP surrogate model are explored by Bastos and O'Hagan (2009). Looking at broader applications than metamodeling, Rasmussen et al.'s (2006) work is a seminal text for practitioners seeking to employ GPs in a machine learning context.

Particularly relevant to the area of model calibration are explorations of the use of GPs to integrate multiple models with varying degrees of fidelity and computational complexity. Qian et al. (2008a) propose a set of Bayesian hierarchical GP models to accommodate a case when low- and high-accuracy data is available. With similar aims, Cumming and Goldstein (2009) describe methods for combining low- and high-accuracy information into a multiscale emulator, as well as providing a design strategy for using the low-fidelity model to select the points at which to evaluate the high-fidelity model. Expanding beyond the paradigm of two levels of fidelity, Goh et al. (2013) use GPs to build a model based on the results of models with several different degrees of fidelity.

Researchers have endeavored to expand the applicability of GP metamodels in numerous ways. To accommodate models that cannot be represented by stationary GPs, Gramacy and Lee (2008) implement a nonstationary GP metamodel using treed partitioning. Qian et al. (2008b) discuss the application of GP metamodels to discrete input spaces. Gratiet et al. (2016) provide a comparative analysis of GPs and

polynomial chaos expansions as metamodels in the context of global sensitivity analysis. Other work has focused on a primary weakness of GPs – their poor scalability for large or high-dimensional data sets. Snelson and Ghahramani (2006) provide an influential method for generating sparse GPs to accommodate large data sets. Liu et al. (2020) review a variety of methods for producing scalable GPs, divided broadly into the categories of global approximations that are based on the full data set, and local approximations that achieve scalability by relying on only a subset of the data at each point in the domain.

5 Optimization

The mathematical underpinnings of model-assisted engineering design are found in the field of optimization. A primary strength of the present work is its success in quantifying uncertainty while undertaking calibration and design. This is an area on which much recent innovation in optimization/design has focused. The influential work of Rockafellar and Wets (1991) describes an algorithm for using a limited set of observations (and hence under uncertainty as to appropriate stochastic model for the system) to solve a multiperiod optimization problem. Sahinidis (2004) offer a comprehensive overview of optimization under uncertainty, analyzing the strengths and weaknesses of a diverse array of approaches. Jin et al. (2003) provide a comparative analysis of a variety of metamodeling techniques with respect to their performance when optimizing under uncertainty. Peherstorfer et al. (2018) offer a survey of meth-

ods for optimization under uncertainty that take advantage of multiple models with varying levels of fidelity and computational expense.

Approaches to optimization and design vary enormously, but most methods can be classed as either gradient-based, evolutionary, or as a form of Bayesian optimization. Ruder (2016) offers a brief but comprehensive overview of gradient descent optimization algorithms. Peitz and Dellnitz (2018) propose a gradient-based approach for multi-objective optimization under uncertainty in which descent directions are chosen so as to account for approximation error in the available information about the gradient of the objective function. This leads one to identify a collection of subsets of the design space which contain the Pareto set for the objective function. Vasilopoulos et al. (2019) demonstrate a method for using gradients to locate an approximate point along the Pareto front and then “trace” the Pareto front to explore it efficiently.

Evolutionary algorithms have become popular methods for optimization in recent times, partly due to their ability to treat the objective function as a black box, without requiring gradient information. This is a feature shared with many forms of BO and with the present work. Jin et al. (2003) provide a survey of work on the use of evolutionary optimization under uncertainty. Zhou et al. (2011) provide a similar survey, focusing on multiobjective evolutionary algorithms. Deb and Gupta (2006) describe two approaches for achieving multi-objective optimization solutions that are robust to small perturbations in the input space.

The present work is a Bayesian approach to optimization problems, and thus

might be considered a form of BO. However, this is misleading, as the term BO is widely used to refer more specifically to a set of techniques loosely following in the footsteps of Jones et al. (1998). In the present work, “BO” refers more specifically to this subset of Bayesian approaches to optimization. Other Bayesian approaches that do not fall within that umbrella include the work of Pelikan et al. (1999) and Pelikan (2005), who describe an application of a Bayesian framework to evolutionary optimization.

Jones et al. (1998) use a GP-based response surface approximation to define an acquisition function for sequential selection of evaluation points of the objective function. Vazquez and Bect (2009), Bect et al. (2012), and Chevalier et al. (2014) extend the approach of Jones, developing a stepwise uncertainty reduction methodology for sequential evaluation of the objective function to minimize resulting uncertainty regarding the optimum. This approach, when applicable (i.e. when it is possible to evaluate the objective function adaptively), can significantly reduce the total number of objective function evaluations required for optimization.

BO’s utility has been enhanced by a number of works aimed at practitioners and researchers seeking to apply BO in diverse areas. Shahriari et al. (2016) introduce the core concepts of BO, and demonstrate its usage in a variety of applications, while Frazier (2018) offers a practical tutorial on the use of BO techniques. Snoek et al. (2012) demonstrate the application of BO to optimization of the hyperparameters of machine learning models. Calandra et al. (2016) provide a demonstrative evaluation

of Bayesian optimization methodologies as applied to optimization under uncertainty in the specific domain of robot locomotion parametrization. Picheny et al. (2019) demonstrate how Bayesian optimization methods can be used to identify Nash equilibria in a game theoretic context.

In addition, many efforts have been made to extend the capabilities of BO and to address its shortcomings. Lizotte (2008) reviews the weaknesses of BO – high computational complexity, poorly understood approximation of the response surface to the objective function, and failure to take advantage of the differentiability of objective functions – and demonstrates how to overcome or mitigate these weaknesses. Letham et al. (2019) develop strategies for applying BO in contexts that suffer from high levels of noise, and Snoek et al. (2015) use neural networks for basis function regression of the objective function in place of GPs, to improve the computational efficiency of BO.

6 Applications

In the present work we adapt the Bayesian model calibration framework stemming from the work of Kennedy and O'Hagan (2001) to apply both to calibration and to model-assisted design. We apply our methodology to two practical applications. One involves the engineering design of a wind turbine blade of fixed outer geometry. The blade is constructed using a composite material, with a fixed choice of matrix and filler materials. Other properties of the composite may be varied to achieve

performance and cost goals for the system. Specifically, we assist in the selection of the *volume fraction* of the material (the ratio of filler to matrix, by volume) as well as the thickness (in millimeters) of the blade material. This sort of material selection would traditionally be performed in an *ad hoc* manner, sacrificing using a list of available materials that are designed separate from this or any particular application. We wed material design with the goals of the engineering design application, so that our search space is not limited to pre-existing or previously studied composites. The engineering goals for our application are the simultaneous minimization of the tip deflection of the blade (in meters) when under load, the twist angle (in radians) of the blade under load, and the cost (in USD) of the composite used for the blade. This application is thus an example of multiobjective optimization.

The other practical application considered in the present work is a vibration isolation design problem. This is a system in which a one mass oscillator is anchored by two leaf springs. An impulse force is applied to the oscillator, inducing vibration in the system. The behavior of the resulting system may be measured via its vertical displacement over time, as well as its amplitude response over frequencies. The finite element model of the system must be calibrated with respect to the elastic modulus of the leaf springs, and the calibrated system must be used for engineering design to minimize resulting vibration.

The following chapters are organized as follows. In chapter two, we adapt the KOH framework and demonstrate its potential as a technique for model-assisted de-

sign. We apply the resulting methodology to the wind turbine blade application. As that system involves multiobjective optimization, it provides an opportunity to demonstrate our method’s ability to explore the Pareto front of a system while providing uncertainty quantification of the resulting estimates. In chapter three, we expand the framework developed in chapter two, in order to effect both calibration and engineering design simultaneously. The ability to perform both of these tasks simultaneously, within a single model and with a single set of experimental observations, is a novel benefit of our approach. We demonstrate the resulting methodology in the vibration isolation design application. Chapter four concludes with a discussion of the results of chapters two and three, and thoughts about future directions for research in this area.

References

- Bartz-Beielstein, T. and Zaefferer, M. (2017). Model-based methods for continuous and discrete global optimization. *Applied Soft Computing*, 55:154–167.
- Bastos, L. S. and O’Hagan, A. (2009). Diagnostics for Gaussian process emulators. *Technometrics*, 51(4):425–438.
- Bayarri, M. J., Berger, J. O., and Cafeo, J. (2007a). Computer model validation with functional output. *The Annals of Statistics*, 35:1874–1906.
- Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J.,

Lin, C.-H., and Tu, J. (2007b). A framework for validation of computer models. *Technometrics*, 49(2):138–154.

Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793.

Bhat, K., Haran, M., and Goes, M. (2010). Computer model calibration with multivariate spatial output. In Chen, M.-H., Dey, D., Müller, P., Sun, D., and Ye, K., editors, *Frontiers of Statistical Decision Making and Bayesian Analysis: in Honor of James O. Berger*, pages 168–184. Springer, New York.

Brynjarsdóttir, J. and O'Hagan, A. (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11).

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1):5–23.

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014). Fast Parallel Kriging-Based Stepwise Uncertainty Reduction With Application to the Identification of an Excursion Set. *Technometrics*, 56(4).

Cox, D. D., Park, J. S., and Singer, C. E. (2001). A statistical method for tuning a computer code to a data base. *Computational Statistics and Data Analysis*, 37(1):77–92.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association*, 96(454):717–729.

Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1997). Pressure Matching for Hydrocarbon Reservoirs: A Case Study in the Use of Bayes Linear Strategies for Large Computer Experiments. pages 37–93. Springer, New York, NY.

Cressie, N. (2015). *Statistics for Spatial Data*. Wiley Series in Probability and Statistics. Wiley.

Cumming, J. A. and Goldstein, M. (2009). Small sample bayesian designs for complex high-dimensional models based on information gained using fast approximations. *Technometrics*, 51(4):377–388.

Curran, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1988). A bayesian approach to the design and analysis of computer experiments. Technical report, Oak Ridge National Lab., TN (USA).

Curran, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963.

Deb, K. and Gupta, H. (2006). Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494.

- Drighei, D. and Mourelatos, Z. P. (2012). Parameter screening in statistical dynamic computer model calibration using global sensitivities. *Journal of Mechanical Design, Transactions of the ASME*, 134(8).
- Frazier, P. I. (2018). A Tutorial on Bayesian Optimization.
- Gattiker, J., Myers, K., Williams, B. J., Higdon, D., Carzolio, M., and Hoegh, A. (2017). *Gaussian Process-Based Sensitivity Analysis and Bayesian Model Calibration with GPMSA*, pages 1867–1907. Springer International Publishing, Cham.
- Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.
- Goh, J., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., and Rutter, E. (2013). Prediction and computer model calibration using outputs from multifidelity simulators. In *Technometrics*, volume 55, pages 501–512. Taylor & Francis Group.
- Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.
- Gratiet, L. L., Marelli, S., and Sudret, B. (2016). Metamodel-based sensitivity analysis: polynomial chaos expansions and gaussian processes. *arXiv preprint arXiv:1606.04273*.

Gu, M. and Wang, L. (2018). Scaled Gaussian stochastic process for computer model calibration and prediction. *SIAM-ASA Journal on Uncertainty Quantification*, 6(4):1555–1583.

Higdon, D., Gattiker, J., Lawrence, E., Jackson, C., Tobis, M., Pratola, M., Habib, S., Heitmann, K., and Price, S. (2013). Computer model calibration using the ensemble kalman filter. In *Technometrics*, volume 55, pages 488–500. Taylor & Francis Group.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466.

Jin, R., Du, X., and Chen, W. (2003). The use of metamodeling techniques for optimization under uncertainty. *Structural and Multidisciplinary Optimization*, 25(2):99–116.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. Technical report.

Kennedy, M. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

- Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464.
- Letham, B., Karrer, B., Ottoni, G., Bakshy, E., et al. (2019). Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519.
- Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2020). When Gaussian Process Meets Big Data: A Review of Scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–19.
- Lizotte, D. J. (2008). *Practical bayesian optimization*. University of Alberta.
- Loeppky, J. L., Bingham, D., and Welch, W. J. (2006). *Computer model calibration or tuning in practice*. Department of Statistics, University of British Columbia.
- Mitchell, T. J. and Morris, M. D. (1992). Bayesian design and analysis of computer experiments: two examples. *Statistica Sinica*, pages 359–379.
- Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632.
- Paulo, R., García-Donato, G., and Palomo, J. (2012). Calibration of computer models with multivariate output. *Computational Statistics and Data Analysis*, 56:3959–3974.

Peherstorfer, B., Willcox, K., and Gunzburger, M. (2018). Survey of multifidelity methods in uncertainty propagation, inference, and optimization.

Peitz, S. and Dellnitz, M. (2018). Gradient-based multiobjective optimization with uncertainties. In *NEO 2016*, pages 159–182. Springer.

Pelikan, M. (2005). Hierarchical Bayesian Optimization Algorithm. pages 105–129. Springer, Berlin, Heidelberg.

Pelikan, M., Goldberg, D. E., and Cantú-Paz, E. (1999). BOA: The Bayesian Optimization Algorithm. In Banzhaf, W., Daida, J., Eiben, A., Garzon, M., Honavar, V., Jakiel, M., and Smith, R., editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO99*, pages 525–532, Orlando, Florida. Morgan-Kaufmann.

Picheny, V., Binois, M., and Habbal, A. (2019). A Bayesian optimization approach to find Nash equilibria. *Journal of Global Optimization*, 73(1):171–192.

Pratola, M. and Chkrebtii, O. (2018). Bayesian calibration of multistate stochastic simulators. *Statistica Sinica*, 28:693–719.

Pratola, M. T., Sain, S. R., Bingham, D., Wiltberger, M., and Rigler, E. J. (2013). Fast sequential computer model calibration of large nonstationary spatial-temporal processes. *Technometrics*, 55(2):232–242.

Qian, P. Z. G., Jeff, C. F., and Stewart, W. H. M. (2008a). Bayesian hierarchical mod-

eling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204.

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008b). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396.

Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. (2006). Building surrogate models based on detailed and approximate simulations. *Journal of Mechanical Design, Transactions of the ASME*, 128(4):668–677.

Rasmussen, C. E., Williams, C. K. I., Sutton, R. S., Barto, A. G., Spirtes, P., Glymour, C., Scheines, R., Schölkopf, B., and Smola, A. J. (2006). *Gaussian processes for machine learning*. MIT Press, Cambridge.

Rockafellar, R. T. and Wets, R. J.-B. (1991). Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Mathematics of Operations Research*, 16(1):119–147.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6-7):971–983.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The design and analysis of computer experiments*. Springer, New York.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Simpson, T., Toropov, V., Balabanov, V., and Viana, F. (2008). Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come—or not. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 5802.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudoinputs. In *Advances in neural information processing systems*, pages 1257–1264.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. (2015). Scalable bayesian optimization using deep

neural networks. In *International conference on machine learning*, pages 2171–2180.

Sorokowska, A., Sorokowski, P., Hilpert, P., et al. (2017). Preferred interpersonal distances: A global comparison. *Journal of Cross-Cultural Psychology*, 48(4):577–592.

Thompson, P. A. and Marchant, E. W. (1995). A computer model for the evacuation of large building populations. *Fire Safety Journal*, 24:131–148.

Trucano, T. G., Swiler, L. P., Igusa, T., Oberkampf, W. L., and Pilch, M. (2006). Calibration, validation, and sensitivity analysis: What’s what. *Reliability Engineering and System Safety*, 91(10-11):1331–1357.

Vasilopoulos, I., Asouti, V. G., Giannakoglou, K. C., and Meyer, M. (2019). Gradient-based pareto front approximation applied to turbomachinery shape optimization. *Engineering with Computers*, pages 1–11.

Vazquez, E. and Bect, J. (2009). A sequential bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes*, 42(10):546–550.

Westermann, P. and Evins, R. (2019). Surrogate modelling for sustainable building design—a review. *Energy and Buildings*, 198:170–186.

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., and Keller-McNulty,

S. (2006). Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis*, 1(4):765–792.

Wong, R. K. W., Storlie, C. B., and Lee, T. C. M. (2014). A frequentist approach to computer model calibration. *Journal of the Royal Statistical Society: Series B*, 79(2):635–648.

Yuan, J. and Ng, S. H. (2013). A sequential approach for stochastic computer model calibration and prediction. *Reliability Engineering and System Safety*, 111:273–286.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.

Multi-objective engineering design via computer model calibration

Carl Ehrett*

Watt Family Innovation Center,
Clemson University,

D. Andrew Brown
School of Mathematical and Statistical Sciences,
Clemson University,

Christopher Kitchens
Department of Chemical and Biomolecular Engineering,
Clemson University,

Evan Chodora
Department of Mechanical Engineering,
Clemson University,

Sez Atamturktur
Department of Architectural Engineering,
Pennsylvania State University

Abstract

Computer model calibration typically operates by fine-tuning parameter values in a computer model so that the model output faithfully predicts reality. By using performance targets in place of observed data, we show that calibration techniques can be repurposed for solving multiobjective design problems. Our

*The authors gratefully acknowledge grant CMMI-1934438 from the National Science Foundation (NSF). CE was supported by fellowships through Department of Education GAANN grant P200A150310 and NSF NRT grant 1633608. DAB is also supported by NSF grants EEC-1744497 and OIA-1826715.

approach allows us to consider all relevant sources of uncertainty as an integral part of the design process. We demonstrate our proposed approach through both simulation and fine-tuning material design settings to meet performance targets for a wind turbine blade.

Nomenclature

c Cost (USD)

C Covariance function of a Gaussian process

\mathbf{C}_D Covariance matrix formed with covariance function C and data D

D concatenated vector of computer model runs and target values ($= (\boldsymbol{\eta}^T, \mathbf{y}_t^T)^T$)

d Tip deflection (m)

h Temperature (Kelvin)

k Thickness (mm)

f Function describing a phenomenon of interest

f_γ Function describing a phenomenon of interest in state γ

m Number of outputs of η

p Dimension of \mathbf{x}

r Twist angle (radians)

\mathbf{t} Vector of model inputs such that the true/optimal value of \mathbf{t} is $\boldsymbol{\theta}$

v Volume fraction of a composite material

\mathbf{x} Array of model inputs other than $\boldsymbol{\theta}$

- y Function relating inputs \mathbf{x} to outputs \mathbf{y}
- \mathbf{y} Vector of model outputs
- \mathbf{y}_t Vector of target model outputs
- \mathbf{z} Vector of outputs of η summed with error ϵ
- α The true state of a system
- β_j^γ Inverse correlation length for j^{th} input of the Gaussian process emulator of γ
- δ Systematic model bias
- ϵ Mean-zero noise
- ζ_γ Smoothness hyperparameter for the Gaussian process emulator of γ
- $\boldsymbol{\theta}$ Optimal inputs to a given model
- η Computer model simulator of f
- $\boldsymbol{\eta}$ Array of outputs from η
- λ_γ Marginal precision of Gaussian process emulator of γ
- μ Mean function of a Gaussian process
- π A probability density function (pdf)
- ρ_j^γ Reparameterization of β_j^γ
- σ^2 Variance of ϵ
- ω A possible system state
- \mathcal{D} The domain of a Gaussian process
- \mathcal{P} Pareto set for multiple objectives

1 Introduction

In the design of engineering systems, multiple performance outcomes are balanced against budgetary constraints. Among the complexities of optimizing over multiple objectives is the effect of uncertainties in the problem. Design is guided by models known to be imperfect, systems are built using materials with partially unknown properties, variations occur in the construction of designed systems, and so on. These imperfections, uncertainties, and errors cause uncertainty also in the solution to a design problem.

In this paper, we cast the engineering design problem in the framework of computer model calibration under uncertainty. In traditional calibration, one aligns computer model output to observations of a real system by estimating unknown parameters in the model. Here, we instead align the computer model to performance and cost targets by finding design variables that optimize the model output with respect to those targets.

Our proposed methodology uses the framework first established by Kennedy and O'Hagan (2001). This area is furthered by Higdon et al. (2004), who undertake a fully Bayesian approach to model calibration. The approach is refined and exemplified by Williams et al. (2006) for a flyer plate experiment. Loeppky et al. (2006) offer a maximum likelihood-based alternative to the Bayesian approach advocated by Kennedy and O'Hagan, intending thereby to improve the identifiability of the calibration parameters in the face of model discrepancy. Bayarri et al. (2007) extend the approach

of Kennedy and O'Hagan, allowing for simultaneous validation and calibration of a computer model. Bayarri et al. (2007) apply this methodology to computer models with functional output using a hierarchical framework for the coefficients of a wavelet representation. Similarly, Paulo et al. (2012) apply the approach of Bayarri et al. (2007) to computer models with multivariate output. Brynjarsdóttir and O'Hagan (2014) demonstrate the importance of strong priors on the model discrepancy term to improve identifiability and interpretability of calibration parameters.

Common to those approaches is a conception of calibration as using real observations to get a posterior distribution on unknown parameters so that the posterior predictive distribution of the model approximates reality. By contrast, using an approach we call counterfactual Bayes, our methodology uses artificial observations (representing design targets) to obtain a posterior distribution on design variables so that the posterior predictive distribution approaches those targets. In counterfactual Bayes, we apply Bayesian reasoning to a hypothetical scenario that bears certain known relationships to reality. Those known relationships allow us to transfer knowledge gained about the hypothetical scenario to reality, thereby gaining valuable insights into the phenomenon of interest. We describe how, with little added computational cost, the methodology provides an initial rough estimate of the *Pareto front* for the system as well as its inverse image in the design space, called the *Pareto set*. (A design point is Pareto optimal if and only if, in order to improve any one of its objectives, some other objective must be made worse off.) This initial rough estimate of the Pareto front

can be used to select artificial observations closer to the design space and thereby promote stronger Bayesian learning about the Pareto set. Repeated applications of the procedure can be used to produce more thorough “Pareto bands” which estimate the Pareto front with quantified uncertainties.

A prominent class of algorithms for multi-objective optimization (MOO) is gradient-based approaches, some of which account for uncertainty. For instance, Peitz and Dellnitz (2018) propose an approach for finding the Pareto set in which descent directions are determined while accounting for approximation error in the gradient information and approximate function evaluations, leading to a collection of subsets of the design space thought to contain the Pareto set. Vasilopoulos et al. (2019) use function gradients to locate an approximate point along the Pareto front, followed by “tracing” the Pareto front to efficiently explore it in a bi-objective optimization problem. Such approaches exploit information about the gradient of the objective function to find optimal directions of descent or exploration. We are concerned here with situations in which the objective function is a “black box” for which the gradient information is unavailable. Peitz and Dellnitz propose a gradient-free version of their approach in which the subsets are found through trial and error in a sampling algorithm. By contrast, we avoid the use of gradients but still inform the direction of exploration by using prior information about what a “good outcome” looks like (i.e., target performance).

Our approach is an example of Bayesian MOO under uncertainty. Concerns about

uncertainty in optimization may include uncertainty in the inputs (as when the inputs are not perfectly known), uncertainty in the outputs (as when the code or process of interest is not deterministic), and observation error (Jin and Branke, 2005; Deb and Gupta, 2006; Zhou et al., 2011).

In traditional Bayesian optimization (BO), a Gaussian process (GP) surrogate model is constructed based on a small set of training observations, and the resulting updated GP is used to define an “acquisition function” that is used sequentially to select new observation locations until a stopping condition is achieved (Picheny et al., 2019). Acquisition functions are crafted to attempt to balance exploration with exploitation of the objective function. Examples include efficient global optimization (Jones et al., 1998) and stepwise uncertainty reduction (Chevalier et al., 2014), the latter of which is applied to MOO by Picheny (2015). Tuo and Wang (2020) provide uniform error bounds for Bayesian global optimization using GPs. Pandita et al. (2018) extend BO to stochastic MOO.

The methodology we propose here differs from these forms of BO by its avoidance of sequential sampling, which is desirable in cases where the computational budget is very small or the data-gathering process is independent of the optimization. Our methodology also can be used to quantify all associated forms of uncertainty discussed above – uncertainty due to the model inputs, due to the stochastic nature of the objective function, or due to observation error of the outputs. Our approach thus has affinities with that of Olalotiti-Lawal and Datta-Gupta (2018), whose approach

captures uncertainty remaining in the distribution designed by the authors. By contrast, under our approach, the distribution explored via Markov chain Monte Carlo (MCMC; Gelfand and Smith, 1990) is dictated by the model itself (and by the GP surrogate thereof), by our prior knowledge about the appropriate design settings, and by the choice of performance/cost targets. Our approach also may be used as a form of “goal programming” (Miettinen, 2008), targeting a particular region of the Pareto front in accordance with design preferences.

Our approach is motivated by the desire to couple material selection and engineering system design under the umbrella of MOO with uncertainty. Material discovery / selection and engineering system design are typically done independently of each other. In particular, we apply our proposed methodology both to a proof-of-concept example and to finding material design settings to optimize performance and cost for a wind turbine blade of fixed outer geometry. The goal is to reduce the twist angle and tip deflection of the blade under load while keeping unit cost of the composite material low.

In Section 2, we describe the counterfactual Bayes methodology for learning about a real system by applying Bayesian reasoning in a hypothetical scenario with known linkages to the real system. In Section 3, we review the calibration framework and how it can be repurposed for design optimization. In Section 4 we apply our methodology to a simulated example with a known truth. We consider the wind turbine blade design problem in Section 5. Section 6 concludes with discussion and thoughts about

future directions.

2 Counterfactual Bayes

Counterfactual Bayes relies on reasoning about counterfactual situations, a cornerstone of causal inference (Rubin, 1974). To elucidate, we rely on the conception of possible states of a system, each of which is internally consistent, but may or may not match the actual system being studied (Adams, 1974; Lewis, 1986). For example, while it is perhaps true that all dogs weigh under 200kg, one can conceive of a world in which some dogs weigh over 200kg, without contradiction; i.e., a 200kg dog *could* exist. By contrast, there is no possible world in which some dogs are reptiles, since dogs are mammals by definition. To describe any creature simultaneously as a reptile and as a dog is a contradiction.

We can summarize the methodology of counterfactual Bayes as follows. Let α denote the true state of a system and f_α a function relating inputs $\mathbf{x}, \boldsymbol{\theta}$ to some output \mathbf{y} , describing some outcome of interest for which we wish to find optimal settings for $\boldsymbol{\theta}$. Suppose that f_α is such that the optimal outcome can be defined in terms of some desired outcome \mathbf{y}_t ; i.e., $\operatorname{argmin}_{\boldsymbol{\theta}} f_\alpha(\mathbf{x}, \boldsymbol{\theta}) = \operatorname{argmin}_{\boldsymbol{\theta}} \|\mathbf{y}_t - f_\alpha(\mathbf{x}, \boldsymbol{\theta})\|$ for some target \mathbf{y}_t and some norm $\|\cdot\|$. Then a distribution $\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}_t$ can be constructed on values producing the optimal achievable output of the system. This notion is similar to using a so-called Gibbs posterior to minimize a given risk function (Jiang and Tanner, 2008). Consider now a possible state ω in which the outcomes are

indistinguishable from those of the true state, $f_\omega = f_\alpha$, and in which we observe \mathbf{y}_t . Then we can apply Bayes' rule to learn a posterior distribution $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}_t)$ of $\boldsymbol{\theta}$ values in ω . While not directly applicable to the true state, we have that $\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}_t$ approximates a distribution on $\boldsymbol{\theta}$ values producing an optimal achievable outcome from the system f_ω and $f_\alpha = f_\omega$. Thus, a distribution on $\boldsymbol{\theta}$ values optimal for f_ω is also a distribution on $\boldsymbol{\theta}$ values optimal for f_α . Thus by relying on known connections between ω and α , we use observations made only assuming state ω to gain valuable insight into features of the true state α .

In what follows, we apply this counterfactual Bayes approach to find distributions on optimal design settings. In our approach, we apply the model calibration framework (Kennedy and O'Hagan, 2001) in a hypothetical scenario involving artificial observations of idealized outcomes \mathbf{y}_t , using our knowledge of the true system to exploit the resulting posterior distribution $\boldsymbol{\theta}|\mathbf{y}_t$, thereby finding a distribution on optimal design settings.

3 Calibration for design

3.1 Gaussian process emulators for calibration

In this work, we use Gaussian processes (GPs) for emulators of computationally expensive computer models. As a multivariate Gaussian random variable is characterized by a mean vector and a covariance matrix, a GP is characterized by mean

and covariance functions $\mu : \mathcal{D} \rightarrow \mathbb{R}$ and $C : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$, where \mathcal{D} is the domain of the process. For points $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\mu(\mathbf{x})$ is the GP mean at \mathbf{x} , and $C(\mathbf{x}, \mathbf{y})$ is the covariance between the values of the GP at \mathbf{x} and \mathbf{y} . The distribution of the GP at any finite number of points is multivariate normal with mean vector and covariance matrix determined by $\mu(\cdot)$ and $C(\cdot, \cdot)$. In principle, model calibration need not rely on emulators; one can complete a Bayesian analysis via MCMC by running the model at each iteration of the chain (Hemez and Atamturktur, 2011). In Section 4 we assume fast-running computer code for the simulated example, but computer models are often too computationally expensive to allow such expenditure (Van Buren et al., 2013, 2014).

The use of GPs as a computationally efficient predictor of computer code given observations of code output is advocated by Sacks et al. (1989) and explored at length by Santner et al. (2003). This is due to a GPs flexibility, interpolating property, and closed-form expressions for uncertainty quantification. Since computer code is typically deterministic (with some exceptions; Pratola and Chkrebtii, 2018), these applications differ from the focus of O'Hagan (1978). Kennedy and O'Hagan (2001) uses GPs for computer model calibration. Kennedy et al. (2006) showcase this use of GP emulators for uncertainty and sensitivity analyses. Bastos and O'Hagan (2009) describe numerical and graphical diagnostic techniques for assessing when a GP emulator is successful, as well as likely causes of poor diagnostic results. Though most work on GP emulation uses stationary covariance functions and quantitative inputs,

Gramacy and Lee (2008) use treed partitioning for a nonstationary computer model, and Qian et al. (2008) explore methods that include both quantitative and qualitative inputs.

Whether or not an emulator is used, one may consider a computer model to be of the form $\eta(\mathbf{x}, \boldsymbol{\theta})$, where $(\mathbf{x}, \boldsymbol{\theta})$ comprise all model inputs. The vector $\boldsymbol{\theta}$ denotes the inputs to be calibrated, and the vector \mathbf{x} denotes *operational domain inputs*, variables for different values of which the design must satisfy the performance expectations. Thus, the model used for calibration (Kennedy and O'Hagan, 2001) is typically taken to be

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon(\mathbf{x}), \quad (1)$$

where $y(\mathbf{x})$ is the observed response at operational domain inputs \mathbf{x} , $f(\cdot)$ is the true system, $\delta(\cdot)$ is the model discrepancy (the systematic bias of the model) and $\epsilon(\cdot)$ is mean-zero observation error, often assumed to be i.i.d. Gaussian.

To use an emulator, suppose we have inputs $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^n \subseteq \mathbb{R}^p \times \mathbb{R}^q$ scaled to the unit hypercube and completed model runs $\eta(\mathbf{x}_i, \mathbf{t}_i)$ for $i = 1, \dots, n$. Define the GP prior for $\eta(\cdot, \cdot)$ as having mean function $\mu(\mathbf{x}, \mathbf{t})$, usually taken to be constant, and set the covariance function in terms of the marginal precision λ_η and a product power

exponential correlation:

$$C((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')) = \frac{1}{\lambda_\eta} \prod_{k=1}^p \exp(-\beta_k^\eta |x_k - x'_k|^{\zeta_\eta}) \times \prod_{j=1}^q \exp(-\beta_{p+j}^\eta |t_j - t'_j|^{\zeta_\eta}) + \sigma^2 I_{(\mathbf{x}, \mathbf{t}) = (\mathbf{x}', \mathbf{t}')}, \quad (2)$$

where β_k , $k = 1, \dots, p + q$, describes the strength of the GP's dependence (i.e., sensitivity) on input direction k , and ζ_η determines the smoothness of the GP (i.e., the differentiability of the sample paths). Independent Gaussian observation error is captured by σ^2 and the indicator I . If $\eta(\cdot, \cdot)$ is a deterministic computer model, then we can set $\sigma^2 = 0$. The model is completed by specifying priors for the hyperparameters $c, \lambda_\eta, \alpha_\eta, \beta_j^\eta$ and σ^2 for $j = 1, \dots, p + q$, though in practice these are often set to predetermined values or estimated from the data via, e.g., maximum likelihood.

3.2 Design to target outcomes

Call design targets treated as observations in the design procedure we propose below “target outcomes”, and call that procedure, which pairs a Bayesian model calibration framework with target outcomes via counterfactual Bayes, “calibration to target outcomes” (CTO). Thus target outcomes are a sort of artificial data, and the calibration procedure is carried out as if these artificial data had been observed in reality. As in traditional calibration, in which the result is a distribution on the calibrated

parameter $\boldsymbol{\theta}$ to approximate the observed data, in CTO the result is a distribution on the design parameter $\boldsymbol{\theta}$ which induces the model to approximate the performance and cost targets. Note that the Bayesian model calibration framework allows for quantification of all sources of uncertainty, including uncertainty about the values of model inputs other than the design variables, uncertainty introduced from using a surrogate in place of the actual computer model, and model form uncertainty (i.e., how closely the code approximates reality).

In the Kennedy-O'Hagan framework, the goal is computer model calibration, so that $\eta(\cdot, \cdot)$ is a computer model representing some real phenomenon $f(\cdot)$. The framework is naturally suited to computer model calibration because $\boldsymbol{\theta}$ is an input for $\eta(\cdot, \cdot)$ but not for the real system of interest $f(\cdot)$. By contrast, in CTO, $\boldsymbol{\theta}$ is an input for the real system of interest, since $\boldsymbol{\theta}$ is a design setting for the system. Thus under CTO we may take $\eta(\cdot, \cdot)$ either to be a computer model as under KOH, or, alternatively, we may take $\eta(\cdot, \cdot)$ itself to be the real system of interest. In either case, a set $\boldsymbol{\eta}$ of observations of $\eta(\cdot, \cdot)$ can be used to produce a GP model. When $\eta(\cdot, \cdot)$ is the real system, there is no discrepancy. If $\eta(\cdot, \cdot)$ is a computer model, the process of calibrating that model takes place separately from CTO, so the known (estimated) discrepancy term can be absorbed into the $\eta(\cdot, \cdot)$ term. Either way, we can take the discrepancy term in (1) to be $\delta(\cdot) \equiv 0$. As a result, CTO is not afflicted by the identifiability concerns of the Kennedy-O'Hagan framework (Bayarri et al., 2007; Tuo and Wu, 2016).

It is common to plug in the maximum likelihood estimates of the GP covariance

hyperparameters λ_η and β^η in (2) instead of including them in a full Bayesian analysis (Kennedy and O'Hagan, 2001; Santner et al., 2003; Qian et al., 2008; Paulo et al., 2012). In our proposed methodology, that is not merely a convenience, but rather is essential to avoid training an emulator using the target outcomes, which by their nature are extreme outliers. See Liu et al. (2009) on the dangers that arise here. We use values found by maximizing the log likelihood of the available simulation runs with respect to λ_η and β^η . We set the GP to have a constant mean, which works well when (as here) responses are centered and standardized, and when the GP is not used for extrapolation (Bayarri et al., 2007). We set $\zeta_\eta = 2$, implicitly assuming that the model output is infinitely differentiable.

Denote completed runs of the simulator $\boldsymbol{\eta} = (\eta(\mathbf{x}_1, \mathbf{t}_1), \dots, \eta(\mathbf{x}_n, \mathbf{t}_n))^T$, target outcomes $\mathbf{y}_t = (y_t(\mathbf{x}_{n+1}), \dots, y_t(\mathbf{x}_{n+m}))^T$, and $D = (\boldsymbol{\eta}^T, \mathbf{y}_t^T)^T$. Following the counterfactual framework, we take the distribution of $D|\boldsymbol{\theta}, \sigma^2, \hat{\lambda}_\eta, \hat{\rho}^\eta$ to be multivariate normal with mean $\mathbf{0}$ and covariance $\mathbf{C}_D = \{C((\mathbf{x}_i, \mathbf{t}_i), (\mathbf{x}_j, \mathbf{t}_j)) + \sigma^2 I_{i=j>n}\}_{i,j=1}^{n+m}$. Here, σ^2 reflects our assumption that in the hypothetical state ω , the performance targets are unattainable, and hence observable only due to observation error. The *observed performances* under the optimal design inputs deviate randomly according to a distribution with variance σ^2 . This “noise” is introduced by measurement error and random, uncontrollable factors that affect performance (e.g., small variations in the manufacturing process or external factors in the operating environment). We can more generally refer this as *unstructured variation* to distinguish it from random be-

havior that has structure, e.g. a smooth sample path realized from a GP. If $\eta(\cdot, \cdot)$ is a stochastic system, then it may be sufficient to set $\sigma^2 = 0$. However, when the specified targets are extreme outliers, such as unattainable “utopia points” discussed below, it is necessary to include σ^2 . This allows for large deviation between the target and the true optimum without making the state of the system contradictory. In other words, it is necessary to construct the model so that the target outcomes are compatible (i.e., that the hypothetical state is self consistent). Including σ^2 ensures that this requirement is satisfied.

When $\eta(\cdot, \cdot)$ has $m > 1$ outputs, it is standard practice to fit a separate, independent GP to each output (Picheny, 2015). We take this approach here, letting σ_i^2 be the variance of the unstructured variation for the i^{th} output. The variance of the unstructured variation can be set *a priori* based on knowledge, or it can be assigned a prior distribution centered at some baseline value. For instance, setting an exponential prior on each σ_i^2 with mean 0.001 corresponds to prior knowledge that that the system has little unstructured variation, whereas $\sigma_i^2 \sim \text{gamma}(4, 1/8)$ induces a heavy-tailed predictive distribution that is more robust to uncontrollable variations in the system performance.

We typically take a uniform prior on the design variables $\boldsymbol{\theta}$ so that it has density $\pi(\boldsymbol{\theta}) \propto 1$. We include also a probability density for operational domain inputs \mathbf{x} , $\pi(\mathbf{x}) \propto 1$, appropriate for systems for which we have additional random, observable inputs that affect the system (e.g., external environmental factors). The joint

posterior density under the model is

$$\pi(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2 | D, \widehat{\lambda}_\eta, \widehat{\boldsymbol{\rho}}^\eta) \propto \pi(D | \boldsymbol{x}, \boldsymbol{\theta}, \widehat{\lambda}_\eta, \widehat{\boldsymbol{\rho}}^\eta) \times \pi(\boldsymbol{\sigma}^2). \quad (3)$$

MCMC methods are used to explore the posterior distribution.

When one has little information about the location and shape of the system’s Pareto front in a multiobjective design problem, it may not be obvious what target best accords with one’s goals. One common choice in such situations is to locate the portion of the Pareto front closest to the “utopia point,” the global minimum of each objective function. When one has access to a set of observations $\boldsymbol{\eta}$, the utopia point can be estimated by taking the minima of the observations of each objective. However, another option in such cases is to perform a “preliminary round” of CTO to estimate the system’s Pareto front. In preliminary CTO, one performs the usual CTO routine with a target known to dominate the utopia point and with σ_i^2 set to a large constant for each objective. By allowing for a large amount of unstructured variation relative to the prior, the prior information dominates the information from the targets in the posterior. This encourages exploration of broad regions of the feasible design space near the Pareto front, since essentially the entire prior support is viewed as compatible with the targets. When the resulting posterior samples of $\boldsymbol{\theta}$ are filtered to retain only their approximate Pareto set, we obtain a rough estimate of the Pareto front that can be used to select target outcomes in an informed way. In addition to being only a rough estimate of the Pareto front, this preliminary estimate

does not include quantification of uncertainties regarding its location. Methods for estimating the system's entire Pareto front with quantified uncertainties are explored in Section 5.4. The full CTO process, including preliminary Pareto front estimation, is given in Algorithm 1.

Algorithm 1: Full CTO procedure including preliminary estimation of Pareto front

1. Set target outcomes \mathbf{y}_t to dominate a known utopia point and $\boldsymbol{\sigma}^2 = s(1, 1, \dots, 1)^T$ for large constant s .
2. Use MCMC to sample $\boldsymbol{\theta}|\mathbf{y}_t$ and thereby the posterior predictive distribution.
3. Filter the predictions to retain only their Pareto optimal values \mathcal{P} .
4. Select new target outcomes \mathbf{y}_t^* using \mathcal{P} as an estimate of the model's Pareto front.
5. Setting $\sigma_i^2 \sim \text{gamma}(4/1/8)$ (for example) for $i = 1, \dots, m$, use MCMC to draw from $\boldsymbol{\theta}|\mathbf{y}_t^*$.

Figure 1 illustrates the benefits of preliminary CTO. Suppose that, prior to undertaking CTO, we know only that the model outputs are positive and the goal is to simultaneously minimize the competing objectives. Then $(0, 0)$ is a natural choice as a target outcome, despite the fact that it is not feasible. The point closest to $(0, 0)$ is unique in the Pareto front solely in being nearest to the origin, and that choice of target outcome was itself driven merely by our ignorance of the feasible design space. By contrast, suppose now that preliminary CTO has supplied us a rough estimate of the Pareto front, empowering us to choose a different target outcome. For instance, $(1.32, 0.065)$ targets a point of diminishing returns in allowing y_1 to increase further in exchange for a reduced y_2 . Note also that when an emulator is used, preliminary CTO can use the same model observations as the subsequent CTO to train the em-

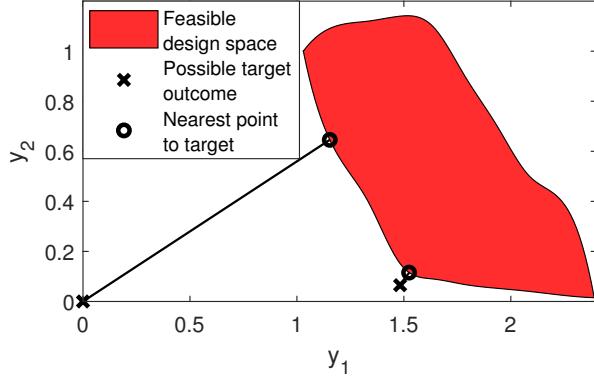


Figure 1: Two choices of target outcomes for CTO, drawing the posterior predictive distribution to two different regions of the feasible design space.

ulator. So preliminary CTO does not add to the budget of model runs, and is thus a computationally cheap supplement to CTO.

4 Simulated Example

To illustrate our proposed procedure, we consider a version of the example problem ZDT1 described by Deb and Sundar (2006). For this illustration we have two objectives y_1, y_2 , and five design variables $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_5)$ with $\theta_i \in [0, 1]$ for all i . We seek optimal settings for $\boldsymbol{\theta}$. Model outputs are $y_1 = \theta_1$ and $y_2 = g(1 - \sqrt{\theta_1/g})$, where $g = 1 + \frac{9}{4} \sum_{i=2}^5 \theta_i$. Though in reality each output is in the range $[0, 1]$, we assume the vague prior knowledge only that the outputs are each in the range $[-6, \infty)$. Figure 2 displays the (normalized) outputs as functions of θ_1 and θ_2 at $x = 2$, where $\theta_i = 0$ for $i = 3, 4, 5$. Assuming an easily evaluated model (so that an emulator is not needed), we have $\mathbf{z}(x) = \eta(\boldsymbol{\theta}) + \boldsymbol{\epsilon}$ for target outcome \mathbf{z} , so that $\boldsymbol{\eta} = (y_1, y_2)^T$ is the output

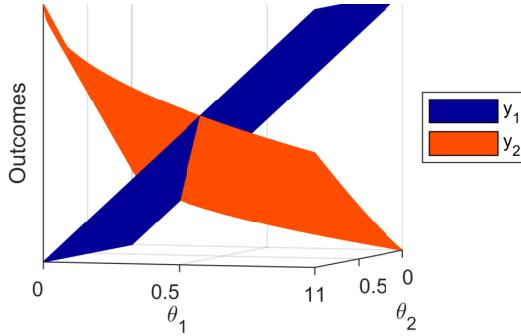


Figure 2: True two-dimensional profile outputs of the five-dimensional simulated example model.

and $\epsilon_i \sim N(\mathbf{0}, \sigma_i^2)$, $i = 1, 2$. For this example we set the prior σ_i^2 to be exponential distributions with mean 0.001 for $i = 1, 2$, corresponding to prior information that there is very little variation in the observed system outputs for a given design setting.

We initially set the target outcomes to $(0.25, -6)$, representing a target chosen with very little knowledge of the location of the system Pareto front. For comparison, we also performed CTO with target $(0.3984, -2.1501)$, which lies much closer to the feasible region (two standard deviations away, under a uniform prior on θ , compared to the original target's 5.2), on the line connecting the original target point to the nearest point in the feasible objective space. Figure 3 shows the resulting posteriors of θ_1 and θ_2 . The marginal posteriors of the remaining inputs are practically indistinguishable from those of θ_2 and thus are not shown. In the top plot, the original target is just over 5.2 units away from the objective space, where each objective is standardized to have variance 1. In the bottom plot, the Euclidean distance of the target from the objective space is 2. The posteriors are similar in the two cases, demonstrating that the method is not sensitive to differences in the distance of the

chosen target from the feasible objective space. The marginals in each case show substantial Bayesian learning compared to the prior (uniform) distribution of the design variables. CTO successfully maps the contours of the optimal region in each case, peaking near the true optimum. This example demonstrates the robustness of CTO to the distance between the target outcome and the feasible objective space. Thus, a target outcome can be selected even when little is known about the location of the Pareto front.

5 Wind turbine material design application

In this section we use CTO to customize a material for use in a wind turbine blade. The material is to be designed specifically for the end use of optimizing blade performance.

5.1 Wind turbine blade design

The two blade performance measures of interest here are tip deflection and twist angle. The engineering design goal is to keep these measures low while also minimizing material cost. The blade is a composite of a given matrix and filler. The material properties (and thus blade performance and cost) depend on the thickness of the shear web in the blade and on the volume fraction, or ratio of filler to matrix. Temperature also affects the composite's properties and hence its performance. It is a known operating condition of the blade but of course is not controllable. Hence, we

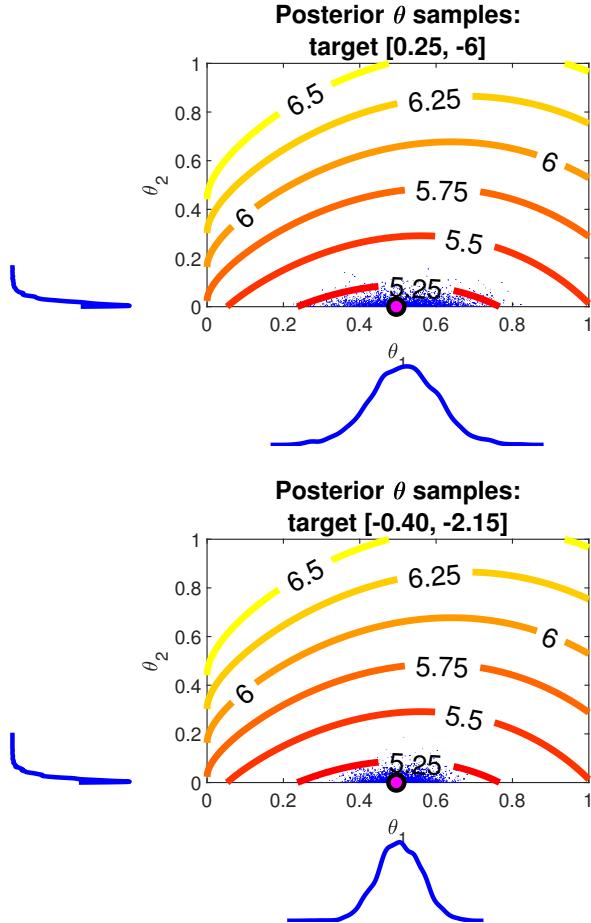


Figure 3: Posterior draws from CTO in the simulated example using an arbitrary (non-feasible) point as a target (top) and using an updated target designed to lie two standard deviations from the Pareto front, in the direction of the original target (bottom). The contours show, for each point in the design space, the Euclidean distance of the model output at that point from the original target point $(0.25, -6)$, when $\theta_i = 0$ for $i = 3, 4, 5$ (which is the optimal setting for those inputs). The large dot shows the true optimum.

treat temperature as an operational domain input but not a design parameter in the computer model. The model inputs are a triplet (h, v, k) , where h is the temperature of the turbine (in kelvin), v is the volume fraction, and k is the thickness (in mm). The model output is a triplet (d, r, c) , where d is tip deflection (in meters), r is twist angle (in radians), and c is cost per square meter (USD) of the material. The turbine is deemed to operate over temperatures 230K-330K.

5.2 Emulation of finite element model

The finite element model is one developed at Sandia National Laboratory for the CX-100 blade. We use ANSYS finite element analysis software (ANSYS, Inc., 2017), interfaced with MATLAB (MATLAB, 2017) code and the NuMAD (Berg and Resor, 2012) manufacturing design tool. The finite element model's estimations of material properties are based on the Mori-Tanaka model (Mori and Tanaka, 1973). Details of the blade and the finite element model may be found in the Appendix. We assume the finite element model accurately represents reality (Van Buren et al., 2013, 2014).

The finite element simulator is too computationally expensive to be suitable for direct use in an MCMC routine. To train the GP emulator, we drew 30 (trivariate) observations from the finite element simulator according to a Latin hypercube sampling design (McKay et al., 1979) based on plausible ranges for the three inputs as identified by subject matter experts: $[230\text{K}, 330\text{K}] \times [0.2, 0.6] \times [10\text{mm}, 25\text{mm}]$. We used a GP with mean 0 and product power exponential covariance function as given

in Equation (2). The GP emulator was validated using 10-fold cross-validation and determined to be an adequate surrogate for the FE model with 30 training points. In fact, there was little difference in predictive ability between 30 and up to 500 training points. Details of the validation of the emulator are in the Appendix.

The hyperparameters λ_η, β^η are estimated via maximum likelihood using only the finite element model output. We used `fmincon()` in MATLAB (MATLAB, 2017) to maximize (with $D = \eta$) over the joint (four-dimensional) support of β^η, λ_η . The estimated values are shown in Table 1.

	d	r	c
$\hat{\rho}_h^\eta$	0.7239	0.7104	1
$\hat{\rho}_v^\eta$	0.9788	0.9723	0.9988
$\hat{\rho}_k^\eta$	0.9906	0.9882	0.9986
λ_η	0.0177	0.0261	0.0009

Table 1: Covariance hyperparameter maximum likelihood estimates for each objective function in the turbine blade example, obtained from 30 training computer runs. For each objective and each input i , $\rho_i^\eta = \exp(-\beta_i^\eta/4)$. The objectives are deflection d , rotation r and cost c .

5.3 Design of the wind turbine blade system

All model inputs were rescaled to [0,1]. All model outputs were standardized so that each of the three responses had mean 0 and standard deviation 1. Initial target outcomes were set to the estimated utopia point (0.6551m, 0.0768rad, \\$96.8) found by taking the minimum observed value of each objective from the 30 simulator observations. The target was replicated to be constant as a function of temperature over an

evenly-spaced grid of temperature values between 230K and 330K.

We carried out preliminary CTO with $\sigma^2 = 5 \times 10^7 \cdot (1, 1, 1)$ to estimate the Pareto front and locate a region of interest. 6,000 iterations were drawn via Metropolis-Hastings-within-Gibbs MCMC (Metropolis et al., 1953; Hastings, 1970; Geman and Geman, 1984) in each of three chains (with random starts), of which the first 3,000 were discarded as burn-in. During the burn-in period, the covariances of the proposal distributions were periodically adjusted to be the sample covariance of the preceding draws scaled for an optimal acceptance rate of around 23% for the multivariate input space (Roberts et al., 1997; Gelman et al., 2013). Convergence of the three chains was verified visually and by the Gelman-Rubin statistic (≈ 1.01 ; Gelman and Rubin, 1992).

As expected for preliminary CTO, the posterior distribution of $\boldsymbol{\theta} = (v, k)$ was quite diffuse. We used the GP emulator to predict the model output for each realization of $\boldsymbol{\theta}$. Figure 4 displays the estimated Pareto front after filtering the posterior predictions to retain only non-dominated performance predictions. Though the objective space is three-dimensional, the Pareto front appears to be a roughly coplanar curve describing a trade-off between cost and deflection/twist. A distinct "knee point" of maximum curvature appears in the Pareto front. This seems to be a point of diminishing returns in the trade-off between performance and cost, and thus we selected this point as the target for design. To do so, we set the point (deflection = 0.743m, twist = 0.089 rad, cost = \$71.11) as the target outcome, replicated to be constant as a

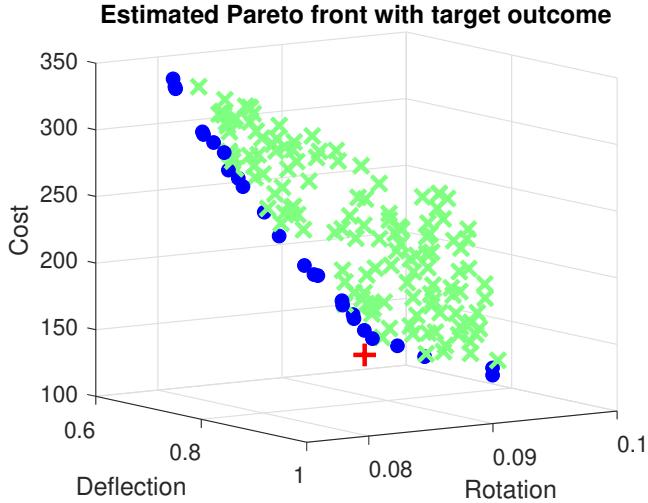


Figure 4: Each x is a dominated design drawn from the predictive distribution through preliminary CTO. The dots indicate the estimated Pareto front. The plus sign is the target selected as the performance objective in our proposed design approach.

function of temperature as in the preliminary round.

In the subsequent CTO, we employed the same MCMC approach as in the preliminary round, except we now assign each element of σ^2 an $\text{Exp}(0.001)$ prior. The covariances of the proposal distributions for each σ_i^2 were periodically adjusted to be the sample covariance of the preceding draws scaled for an optimal acceptance rate of around 44% for the scalar σ_i^2 (Roberts et al., 1997; Gelman et al., 2013). The posterior distribution of $\boldsymbol{\theta}$ appears in Figure 5, with a mode near $(0.6, 10\text{mm})$. Indeed, from the analysis discussed in Section 5.4, we find that the “knee point” in the Pareto front is precisely the point at which volume fraction has reached its upper limit at 0.6, with further gains possible only by raising thickness from its lower limit of 10mm. The contrast of the posterior distribution with the prior, which is uniform over $[0.2, 0.6] \times [10, 25]$, indicates that strong Bayesian learning has occurred. The

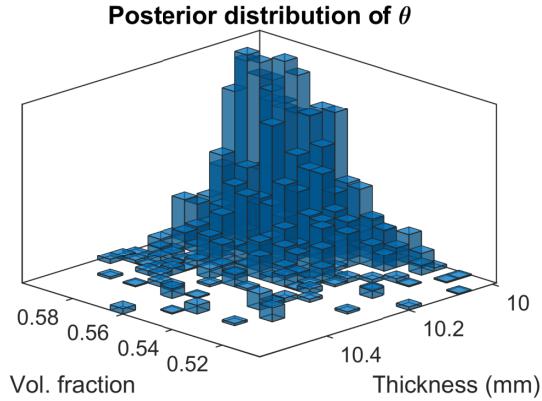


Figure 5: Histogram showing the posterior distribution from CTO in the wind turbine blade system. The prior is uniform over $[0.1, 0.6] \times [10, 25]$.

prior and posterior predictive distributions of the model outputs appear in Figure 6, where the prior predictive distributions are based on a uniform sampling of the model inputs. The mean output under the prior is $(0.753\text{m}, 0.091 \text{ rads}, \$206.58/\text{m}^2)$,

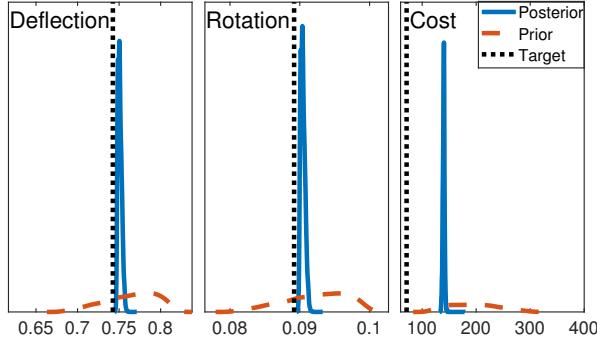


Figure 6: Approximate prior and posterior marginal predictive densities for each of the three outputs in the turbine blade design problem.

and under the posterior it is $(0.751\text{m}, 0.090 \text{ rad}, \$139.80/\text{m}^2)$. Though the mean performance outcomes are approximately the same under the posterior and the prior, mean cost per square meter and the uncertainty of the outcomes are dramatically lower. If one prefers to prioritize gains in performance over cost, this can be accom-

plished by selecting target outcomes that reflect those priorities.

5.4 Pareto front estimation with quantified uncertainties

When multiple design outputs are to be minimized, any point in the Pareto front is optimal relative to some set of priorities. If those priorities have not been explicitly determined prior to the design process, then no particular outcome can be targeted. For example, in a system where performance is monotonically increasing in cost, depending on one's tolerance for high cost, any point in the design space might be optimal. In low-dimensional cases, CTO may be used to achieve a holistic picture of the Pareto front by optimizing to each target outcome on a grid. To do this, where the model output is b -dimensional, one may draw a grid over the range of $b - 1$ of the model outputs and perform CTO to minimize the remaining output at each point of the grid. The $b - 1$ outputs, at each grid point, are treated as known up to small error (e.g., one tenth of one standard deviation from the mean). Allowing some small observation error is necessary because any set of solutions having Lebesgue measure zero has probability zero of occurring. The resulting estimate of the Pareto front differs from the filtering method employed in preliminary CTO in that it allows for quantifying the uncertainty associated with the Pareto front.

Our proposed procedure is illustrated here using the wind turbine blade application. For ease of exposition, twist has been removed as a model output, leaving a system with two-dimensional output of deflection and cost. The range of observed

costs is $[\$96, \$286]$. A 20-point grid was drawn over this range of costs. For each point c in the cost grid, we used the point $(0\text{m}, \$c)$ as the target outcome for calibration (again replicated as constant with respect to temperature). The result is an estimate of the response surface with quantified uncertainty describing, for each point in the grid, the minimal achievable outcome for the output not included in the grid.

The result of applying this strategy to the wind turbine blade application is shown in Figure 7. For comparison, we also plot the results from applying the NSGA-II al-

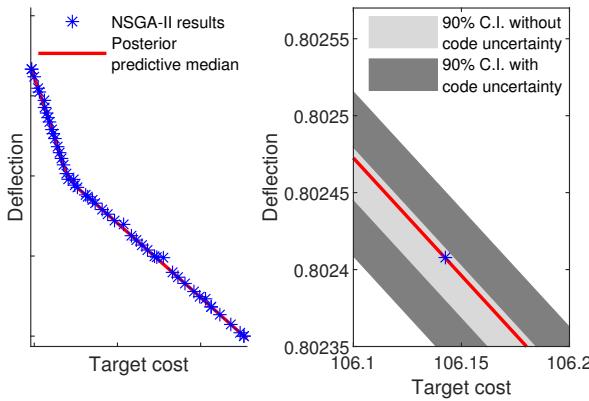


Figure 7: The estimated Pareto front of the wind turbine blade system with quantified uncertainties, along with NSGA-II estimation of the front. The light gray shows the 90% credible interval for the front without code uncertainty (i.e., treating the emulator as perfect); the dark gray extends the credible interval to include code uncertainty.

gorithm (Deb et al., 2002), a popular gradient-free genetic algorithm for MOO. It uses the trained GP emulator as the objective function, with 500 generations and population size 50. NSGA-II and our approach give very similar estimates of the Pareto front's location. On a machine with an Intel Core i7-9750H CPU and 16GB of RAM, NSGA-II required 132 seconds. While our method required more computation time (461 seconds), it generated far more informative results. In contrast with that of

NSGA-II, our approach can quantify all the sources of uncertainty. Such uncertainty is important to account for since no emulator is *identical* to the computer model output, and because uncontrollable factors can affect performance (e.g., uncontrollable operating temperature that changes day to day).

Figure 8 shows the application of CTO to three different problems with a training set of 100 FE model runs each. All three cases attempt bivariate minimization of

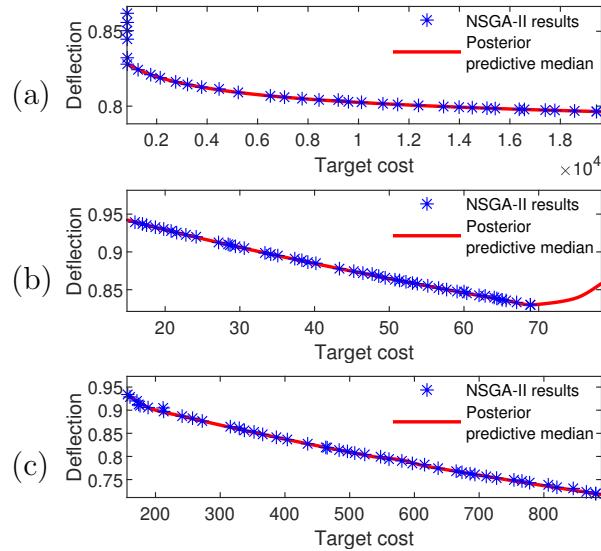


Figure 8: Estimated Pareto front for multiple wind turbine blade systems with respect to a variety of design spaces, along with NSGA-II estimation of the fronts. 95% credible intervals are too small to be visible.

both blade deflection and cost. Subplot (a) searches for optimal design settings for the composite material's filler modulus and matrix modulus. Subplot (b) searches for optimal aspect ratio and shear web thickness. Subplot (c) shows the results for a search over three design variables: aspect ratio, volume fraction and shear web thickness. In each case, CTO is consistent with the results from NSGA-II. Notice also in Subplot (b) that CTO, in contrast with NSGA-II, is informative about the

behavior of the system at costs beyond those in the Pareto front.

The use of CTO in this case demonstrates the value of obtaining a posterior distribution on the design variables, rather than just a point estimate. For example, Figure 5 shows not just that a reasonable point estimate of the optimal θ is at (0.6, 10mm)—respectively the upper and lower extrema of the supports for volume fraction and thickness. We also have information about the variation in the design space corresponding to variation in the observed performance from one experiment to the next. This is potentially useful for studying system tolerances.

The wind turbine case illustrates how our proposed method can deliver “Pareto bands,” providing not merely an estimate of the Pareto front (as in preliminary CTO) but also uncertainty associated with that estimate. Such an estimate can be of use to decision-makers when deciding on performance goals subject to budgetary constraints while also accounting for uncontrollable factors in the manufacturing process or operating environment.

6 Discussion

We have described how the computer model calibration framework of Kennedy and O’Hagan (2001) can be adapted for engineering design. Calibration to target outcomes undertakes design by “calibrating” a model not to field observations, but rather to performance and cost targets. The procedure optionally includes a computationally cheap preliminary step that provides a rough estimate of the Pareto front, which

may be used to select target outcomes that promote strong Bayesian learning. The resulting posterior predictive distribution approximates the target outcomes, so that the posterior distribution of θ constitutes a distribution on optimal design settings. Repeated applications of this methodology allows one to construct a thorough estimate of the Pareto front of the system with quantified uncertainties by selecting target outcomes that explore different portions of the Pareto front.

Unlike other methods of Bayesian optimization (a review of which is provided by Shahriari et al., 2016), CTO does not require the ability to evaluate model output adaptively. Instead, it can rely on a batch of observations gathered prior to (and independently of) the design process. We described the implementation of this approach in an MCMC routine along with considerations to accommodate computational instability. The use of this methodology is illustrated in the case of material design for a wind turbine blade. By expropriating established tools of model calibration, CTO offers a method of optimization which is sensitive to, and quantifies, all sources of uncertainty.

The example of Section 4 has five design inputs and bivariate objectives, and the applications in Section 5 each had either two or three design inputs and two or three objectives. The number of objectives is mostly for ease of illustration and visualization, as well as practical interest in the turbine blade design problem. There exist many design problems with many more objectives and/or design dimensions than those considered here. In considering the computational burden associated with

a larger number of objectives, we follow standard practice by assuming independent Gaussian processes for each output (Picheny, 2015), meaning that the computation essentially scales linearly with the number of outputs. For cases in which independent GPs are not appropriate, it is certainly possible to account for the dependence of the outputs in the surrogate model (Conti and O'Hagan, 2010). The computational burden of such an approach would be more severe in such a case. While CTO with a *single* target can be applied with larger numbers of objectives, the grid-based Pareto front estimation can become prohibitive since the required grid grows exponentially with the dimension of the objective space. With respect to the dimension of the design space, the limitations of CTO here are those that arise from the underlying MCMC algorithm. High-dimensional MCMC is the subject of ongoing research, some of which is reviewed by Saibaba et al. (2019). While an exploration of this issue is beyond the scope of the current work, we remark that marginalization and Hamiltonian Monte Carlo have been shown to be effective. Another partial remedy for these difficulties would be to perform an *a priori* sensitivity analysis in order to reduce the inputs only to those that substantially affect the output. Similarly, one could use active subspaces (Constantine, 2015) to reduce the dimensionality of the design space.

It is possible for there to be proper subsets of the design space that are not feasible (e.g., design values that cannot be meshed for shape optimization), or that are poorly identified by the performance criteria (i.e., an ill-posed inverse problem). The Bayesian approach that we use here is naturally suited for such situations. For

example, the prior on the design space can place zero probability on infeasible subsets or otherwise impose regularization to constrain the space of possible solutions. This latter feature is one of the reasons the Bayesian approach to inverse problems has been gaining popularity over the last few years (Calvetti et al., 2014).

The example and applications we describe here correspond to unconstrained problems. However, methods are available for constructing GPs that incorporate known constraints. For example, Golchi et al. (2015) use sequential Monte Carlo to simulate GPs that are monotone with respect to some or all inputs. Wang and Berger (2016) similarly discuss methods for incorporating shape constraints (including monotonicity) into a GP. Maatouk and Bay (2017) use a functional decomposition to create a finite-dimensional approximation of a GP that allows one to incorporate inequality constraints. Ding et al. (2019) allow for boundary constraints in GP emulation with a mean function that honors the information along with covariance functions that go to zero at the known boundaries. We suspect that it would be straightforward to incorporate such procedures into our proposed design approach.

The methodology as described here treats the computer model as universally valid over the domain of the design variables. Future work in this area will include the use of a discrepancy term capturing model bias. Other possible extensions of our proposed methodology include its application to so-called “state-aware calibration” (Atamturktur and Brown, 2015; Stevens et al., 2018; Brown and Atamturktur, 2018), which would allow the optimal region of the design variables to vary as a function of

the operational domain inputs.

References

- Adams, R. M. (1974). Theories of Actuality. *Noûs* 8(3), 211.
- ANSYS, Inc. (2017). Ansys[®] academic research mechanical, release 18.1.
- Atamturktur, S. and D. A. Brown (2015). State-aware calibration for inferring systematic bias in computer models of complex systems. *NAFEMS World Congress Proceedings, June 21-24*.
- Bastos, L. S. and A. O'Hagan (2009). Diagnostics for Gaussian process emulators. *Technometrics* 51(4), 425–438.
- Bayarri, M. J., J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh (2007). Computer model validation with functional output. *The Annals of Statistics* 35, 1874–1906.
- Bayarri, M. J., J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu (2007). A framework for validation of computer models. *Technometrics* 49(2), 138–154.
- Berg, J. C. and B. R. Resor (2012). Numerical manufacturing and design tool (NUMAD v2.0) for wind turbine blades: User's guide. Sandia National Laboratories Report SAND2012-7028.

Berry, D. S. (2008). Blade System Design Studies Phase II: Final Project Report.

Sandia National Laboratories Report SAND2008-4648.

Berry, D. S. and T. Ashwill (2007). Design of 9-Meter Carbon-Fiberglass Prototype

Blades: CX-100 and TX-100. Sandia National Laboratories Report SAND2007-0201.

Brown, D. A. and S. Atamturktur (2018). Nonparametric functional calibration of computer models. *Statistica Sinica* 28, 721–742.

Brynjarsdóttir, J. and A. O'Hagan (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems* 30(11).

Calvetti, D., J. P. Kaipio, and E. Somersalo (2014). Inverse problems in the Bayesian framework. *Inverse Problems* 30(11).

Chevalier, C., J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet (2014). Fast Parallel Kriging-Based Stepwise Uncertainty Reduction With Application to the Identification of an Excursion Set. *Technometrics* 56(4).

Constantine, P. G. (2015). *Active subspaces : emerging ideas for dimension reduction in parameter studies*. Philadelphia: SIAM.

Conti, S. and A. O'Hagan (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference* 140(3), 640–651.

Deb, K. and H. Gupta (2006, dec). Introducing robustness in multi-objective optimization. *Evolutionary Computation* 14(4), 463–494.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197.

Deb, K. and J. Sundar (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06*, New York, New York, USA, pp. 635. ACM Press.

Ding, L., S. Mak, and C. F. J. Wu (2019). BdryGP: A new Gaussian process model for incorporating boundary information. arXiv preprint 1908.08868.

Gelfand, A. E. and A. F. M. Smith (1990, jun). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85(410), 398–409.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013). *Bayesian data analysis* (3rd ed.). London: CRC Press.

Gelman, A. and D. B. Rubin (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4), 457–472.

Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and

the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6), 721–741.

Golchi, S., D. R. Bingham, H. Chipman, and D. A. Campbell (2015, may). Monotone emulation of computer experiments. *SIAM-ASA Journal on Uncertainty Quantification* 3(1), 370–392.

Gramacy, R. B. and H. K. H. Lee (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* 103(483), 1119–1130.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1), 97–109.

Hemez, F. and S. Atamturktur (2011). The dangers of sparse sampling for the quantification of margin and uncertainty. *Reliability Engineering & System Safety* 96(9), 1220–1231.

Higdon, D., M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing* 26(2), 448–466.

Jiang, W. and M. A. Tanner (2008). Gibbs posterior for variable selection in high-dimensional classification and data mining. *The Annals of Statistics* 36(5), 2207–2231.

- Jin, Y. and J. Branke (2005). Evolutionary optimization in uncertain environments - A survey. *IEEE Transactions on Evolutionary Computation* 9(3), 303–317.
- Jones, D. R., M. Schonlau, and W. J. Welch (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13(4), 455–492.
- Kennedy, M. C., C. W. Anderson, S. Conti, and A. O'Hagan (2006). Case studies in Gaussian process modelling of computer codes. *Reliability Engineering & System Safety* 91(10-11), 1301–1309.
- Kennedy, M. C. and A. O'Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B* 63(3), 425–464.
- Lewis, D. K. (1986). *On the plurality of worlds*. Blackwell.
- Liu, F., M. J. Bayarri, and J. O. Berger (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis* 4(1), 119–150.
- Loeppky, J. L., D. Bingham, and W. J. Welch (2006). *Computer model calibration or tuning in practice*. University of British Columbia: Department of Statistics.
- Maatouk, H. and X. Bay (2017, jul). Gaussian Process Emulators for Computer Experiments with Inequality Constraints. *Mathematical Geosciences* 49(5), 557–582.
- MATLAB (2017). *Version 9.2.0 (R2017a)*. Natick, Massachusetts: The MathWorks, Inc.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2), 239–245.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21(6), 1087–1092.

Miettinen, K. (2008). *Introduction to Multiobjective Optimization: Noninteractive Approaches*, pp. 1–26. Berlin, Heidelberg: Springer Berlin Heidelberg.

Mori, T. and K. Tanaka (1973). Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metallurgica* 21(5), 571–574.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B* 40(1), 1–42.

Olalotiti-Lawal, F. and A. Datta-Gupta (2018). A multiobjective markov chain monte carlo approach for history matching and uncertainty quantification. *Journal of Petroleum Science and Engineering* 166, 759–777.

Pandita, P., I. Bilionis, J. Panchal, B. P. Gautham, A. Joshi, and P. Zagade (2018). Stochastic multiobjective optimization on a budget: Application to multipass wire drawing with quantified uncertainties. *International Journal for Uncertainty Quantification* 8(3), 233–249.

Paulo, R., G. García-Donato, and J. Palomo (2012). Calibration of computer models with multivariate output. *Computational Statistics and Data Analysis* 56, 3959–3974.

Peitz, S. and M. Dellnitz (2018). *Gradient-Based Multiobjective Optimization with Uncertainties*, pp. 159–182. Cham: Springer International Publishing.

Picheny, V. (2015). Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing* 25(6), 1265–1280.

Picheny, V., M. Binois, and A. Habbal (2019, jan). A Bayesian optimization approach to find Nash equilibria. *Journal of Global Optimization* 73(1), 171–192.

Pratola, M. and O. Chkrebtii (2018). Bayesian calibration of multistate stochastic simulators. *Statistica Sinica* 28, 693–719.

Qian, P. Z. G., H. Wu, and C. F. J. Wu (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* 50(3), 383–396.

Resor, B. R. and J. Paquette (2012). A NuMAD model of the Sandia CX-100 Blade. Sandia National Laboratories Report SAND2012-9273.

Roberts, G., A. Gelman, and W. Gilks (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability* 7(1), 120.

- Rubin, D. B. (1974, oct). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66(5), 688–701.
- Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Statistical Science* 4(4), 409–423.
- Saibaba, A. K., J. Bardsley, D. A. Brown, and A. Alexanderian (2019). Efficient marginalization-based MCMC methods for hierarchical Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification* 7(3), 1105–1131.
- Santner, T. J., B. J. Williams, and W. I. Notz (2003). *The design and analysis of computer experiments*. New York: Springer.
- Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE* 104(1), 148–175.
- Stevens, G. N., S. Atamturktur, D. A. Brown, B. J. Williams, and C. Unal (2018). Statistical inference of empirical constituents in partitioned analysis from integral-effect experiments. *Engineering Computations* 35(2), 672–691.
- Tuo, R. and W. Wang (2020). Uncertainty quantification for bayesian optimization. arXiv preprint 2002.01569.
- Tuo, R. and C. F. J. Wu (2016). A theoretical framework for calibration in com-

puter models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification* 4(1), 767–795.

Van Buren, K. L., S. Atamturktur, and F. M. Hemez (2014). Model selection through robustness and fidelity criteria: Modeling the dynamics of the CX-100 wind turbine blade. *Mechanical Systems and Signal Processing* 43(1-2), 246–259.

Van Buren, K. L., M. G. Mollineaux, F. M. Hemez, and S. Atamturktur (2013). Simulating the dynamics of wind turbine blades: Part II, model validation and uncertainty quantification. *Wind Energy* 16(5), 741–758.

Vasilopoulos, I., V. G. Asouti, K. C. Giannakoglou, and M. Meyer (2019, aug). Gradient-based Pareto front approximation applied to turbomachinery shape optimization. *Engineering with Computers*, 1–11.

Wang, X. and J. O. Berger (2016, jan). Estimating shape constrained functions using Gaussian processes. *SIAM-ASA Journal on Uncertainty Quantification* 4(1), 1–25.

Williams, B., D. Higdon, J. Gattiker, L. Moore, M. McKay, and S. Keller-McNulty (2006). Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis* 1(4), 765–792.

Zhou, A., B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang (2011, mar). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1(1), 32–49.

Appendix

Turbine blade finite element model

The wind turbine blade model used in this paper is based on a nine-meter research and development blade developed by Sandia National Laboratory known as the CX-100 (Berry, 2008; Berry and Ashwill, 2007). The purpose of the CX-100 blade is to provide an inexpensive test platform for structural modeling and strength testing and is comprised of a unidirectional carbon-fiber laminate with a fiberglass skin. Using the airfoil geometry and composite layer specifications described in the CX-100 development reports, the geometry of the blade is created using the NuMAD (“Numerical Manufacturing and Design”) tool created by Sandia National Laboratories (Berg and Resor, 2012; Resor and Paquette, 2012). The NuMAD software serves to create the blade geometry based on input airfoil geometry data. Components of the blade (edges, root, spar caps, and shear web) are assigned composite material properties and geometry (layer orientation, quantity, and thickness). See Figure 9.

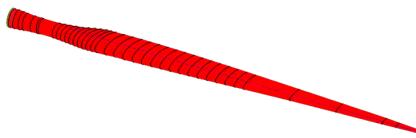


Figure 9: CX-100 blade model created in NuMAD for ANSYS input file generation and finite-element analysis.

The NuMAD software is then used to export the created blade model as an AN-

SYS input file to create the geometry, mesh the body with the appropriate material properties and geometries, and apply the boundary conditions. The model is composed of 8-node structural SHELL281 elements in layers to represent the composite material layers and that support the application anisotropic material properties. The study uses fixed-free boundary conditions where the root is simulated to be fixed to the turbine hub and the tip is free to measure deflection due to loading. Loading is applied to the blade tip as a 6,000N load in the flapwise direction based on measured turbine hub moments of the same blade design under high wind loads of approximately 54,000 N-m. The ANSYS input is modified accordingly to apply the loading, solve the model, and export the nodal displacements and rotations.

Surrogate model validation

The validation of the GP emulator was performed using 10-fold cross validation. Figure 10 shows the results. Though we had access to a large set of 500 finite element model observations, we find that our GP emulator worked well for much smaller training sizes, as shown here. In each case, the RMSE for each of the three outputs is included. Error bars for each observations are also included, but are generally too small to be visible. In each case, we see excellent agreement between the predicted and observed outputs. This validation demonstrates the diminishing returns of using more than 30 finite element model observations to train the emulator.

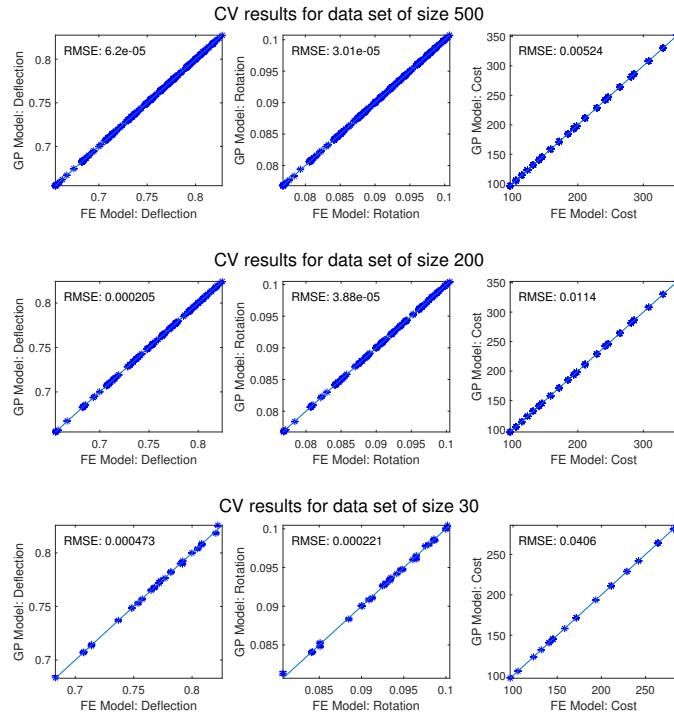


Figure 10: Results of 10-fold cross validation of the GP emulator used for the wind turbine application.

A unified framework for computer model calibration and engineering design

Carl Ehrett*

Watt Family Innovation Center,
Clemson University,

D. Andrew Brown
School of Mathematical and Statistical Sciences,
Clemson University,

Christopher Kitchens
Department of Chemical and Biomolecular Engineering,
Clemson University,

Sez Atamturktur
Department of Architectural Engineering,
Pennsylvania State University,

Xiyue Xu
Department of Architectural Engineering,
Pennsylvania State University, and

Roland Platz
Department of Architectural Engineering,
Pennsylvania State University

Abstract

Calibration of computer models and the use of those models for design are two activities traditionally carried out separately. This paper generalizes ex-

*The authors gratefully acknowledge grant CMMI-1934438 from the National Science Foundation (NSF). CE was supported by fellowships through Department of Education GAANN grant P200A150310 and NSF NRT grant 1633608. DAB is also supported by NSF grants EEC-1744497 and OIA-1826715.

existing Bayesian inverse analysis approaches for computer model calibration to present a methodology combining calibration and design in a unified Bayesian framework. This provides a computationally efficient means to undertake both tasks while quantifying all relevant sources of uncertainty. Specifically, compared with the traditional approach of design using parameter estimates from previously completed model calibration, this generalized framework inherently includes uncertainty from the calibration process in the design procedure. We demonstrate our approach on the design of a vibration isolation system. We also demonstrate how, when adaptive sampling of the phenomenon of interest is possible, the proposed framework may select new sampling locations using both available real observations and the computer model. This is especially useful when a misspecified model fails to reflect that the calibration parameter is functionally dependent upon the design inputs to be optimized.

Keywords: Gaussian processes, optimization, uncertainty quantification, computer model calibration, engineering design

Nomenclature

β Inverse correlation length for Gaussian process input

$\delta(\cdot)$ Discrepancy between model and true system

ϵ_c Measurement error

ϵ_d Discrepancy between optimal system output and \mathbf{y}_t

- $\eta(\cdot)$ Computer model of the system of interest
- ζ Damping ratio of a dynamic vibration system
- $\boldsymbol{\eta}$ Vector of outputs of $\eta(\cdot)$
- $\boldsymbol{\theta}_c$ True value of parameter to be calibrated
- $\boldsymbol{\theta}_d$ Optimal design input
- λ Marginal precision of Gaussian process
- ρ Reparameterization of β
- σ_c^2 Variance of ϵ_c
- σ_d^2 Variance of ϵ_d
- $\boldsymbol{\phi}_\delta$ Hyperparameters of Gaussian process model of $\delta(\cdot)$
- $\boldsymbol{\phi}_\eta$ Hyperparameters of Gaussian process surrogate for $\eta(\cdot)$
- \mathcal{D} $(\boldsymbol{\eta}^T, \mathbf{y}^T)^T$ for some observations \mathbf{y}
- $C_\eta(\cdot)$ Covariance of Gaussian process emulator of η
- $C_\delta(\cdot)$ Covariance of Gaussian process model of δ
- k Elastic modulus of leaf spring
- $f(\cdot)$ The system of interest
- g Gain of dynamic vibration system
- m Mass of oscillator in dynamic vibration system
- $m_\eta(\cdot)$ Mean of Gaussian process emulator of η
- $m_\delta(\cdot)$ Mean of Gaussian process model of δ
- T Time period of one oscillation of a dynamic vibration system

t_c Value of calibration parameter used as input in $\eta(\cdot)$

t_d Value of design variable used as input in $\eta(\cdot)$

\mathbf{x} All model inputs in the operational domain of $f(\cdot)$

\mathbf{y}_r Vector of observations of the system of interest

\mathbf{y}_s Vector of outputs of the computer model of $f(\cdot)$

\mathbf{y}_t Vector of target outcomes for the system of interest

\mathbf{z} All known and/or controllable inputs of $f(\cdot)$

1 Introduction

This paper connects two distinct areas of research concerning computer models of real phenomena. One area is that of computer model calibration, where the goal is to find a posterior distribution on unknown, or imperfectly known, parameters by calibrating a computer model using real-world observations of the modeled phenomenon. The second area is that of enlisting a computer model for design, using the model to find settings for controllable system inputs such that the resulting system output is optimized with respect to some design goal. These two problems are structurally similar, both involving finding estimates or distributions of model inputs to achieve some desired effect on model outputs. In the case of calibration, the desired effect is that the model outputs approximate reality, and in the case of design, the desired effect is that the model outputs approximate the optimal achievable outputs. Since calibration and design are typically carried out separately, existing design techniques

operate under the assumption that the model is an accurate approximation of the real system of interest. In practice, models used for design typically are known or suspected to be biased representations of the phenomenon of interest, and often have inputs that require calibration. The goal of the work described here is to provide a unified framework for calibration and design. We refer to this new approach as DCTO, for dual calibration to target outcomes. In addition to avoiding the idealization that the model used for design is unbiased, DCTO allows one to focus calibration efforts on regions of interest, prioritizing them over other areas of the model range. For example, one may be more interested in calibrating the model to be accurate in the optimal region of some design variable $\boldsymbol{\theta}_d$ than elsewhere. Having a combined framework for calibration and design is especially of interest when those two activities are non-trivially intertwined, as in the case when the value of the calibration parameters are functionally dependent upon the design settings.

Bayesian methods for computer model calibration are developed by Kennedy and O'Hagan (2001). Since their seminal paper, the methodology has seen numerous extensions and refinements (Higdon et al. 2004; Williams et al. 2006; Bayarri et al. 2007a; Bayarri et al. 2007b; Paulo et al. 2012; Brynjarsdóttir and O'Hagan 2014). Henceforth, we refer to this approach to calibration as KOH. Common to KOH approaches is the Bayesian framework in which one places a prior on the calibration parameters $\boldsymbol{\theta}_c$, often pairing it with a Gaussian process (GP) meta-model of the computer model of interest and a GP prior on the model discrepancy $\delta(\cdot)$, and using the

available observations y_r of the real system to find a posterior distribution $\theta_c, \delta(\cdot) | y_r$.

Such an approach is notable for providing not merely a point estimate of the calibration parameter, but for providing a full posterior distribution quantifying remaining uncertainty about θ_c and about $\delta(\cdot)$.

Herein, we leverage the KOH framework to find a posterior distribution, not only on unknown model parameters, but also on controllable design settings. We achieve this via an approach called counterfactual Bayes. In traditional model calibration, one uses Bayes' rule to discover a posterior distribution on calibration parameters using real observations, so that the observations are the source of the Bayesian learning. In a design case, there are no relevant observations. One wants to find design settings that induce the system to behave optimally, but one typically has not observed the system doing so, and therefore there seems to be no relevant source of Bayesian learning that could drive the use of Bayes' rule to discover a posterior distribution on optimal design settings. The idea of counterfactual Bayes is to identify artificial observations, or target outcomes, y_t such that the resulting likelihood is highest in the optimal design region — i.e., target outcomes y_t such that their occurrence is strong evidence that the design settings are optimal. Hence, in addition to calibrating the unknown model parameters against experimental observations, one uses the KOH framework to also find a posterior distribution on design settings given the target outcomes. Given the nature of y_t , this is *de facto* a distribution on optimal design settings for the system. The result retains the benefits of the Bayesian model calibration tools on

which it is based, namely the quantification of remaining uncertainty regarding the optimal design settings. And like KOH, DCTO is especially well-suited to problems that rely on black-box functions.

We may divide optimization approaches in such cases broadly into three camps (Regis and Shoemaker, 2004). Gradient-based approaches (Nocedal and Wright, 2006) are of limited utility when dealing with black-box functions, where we cannot evaluate the objective function's derivative. Approximation of the derivative requires additional function evaluations, rapidly inflating the computational cost when each evaluation involves significant expense. Heuristic approaches (Lee and El-Sharkawi, 2007) such as evolutionary algorithms (Branke et al., 2008; Deb et al., 2002; Kim et al., 2004), particle swarm optimization (Bonyadi and Michalewicz, 2017; Mason et al., 2017), and simulated annealing (Robert and Casella, 2004) avoid the need to know or approximate derivatives, but often require prohibitively many function evaluations. Furthermore, such methods, like gradient-based approaches, do not inherently provide quantification of remaining uncertainty about optimal design settings and the system outputs at those settings. Methods exist for using heuristic approaches while accommodating and quantifying uncertainties (Deb and Gupta, 2006; Zhou et al., 2011), but these come at the cost of even further inflating the number of function evaluations required. This problem can be mitigated by relying on a surrogate model, but the resulting uncertainty quantification is accomplished by separate methods that are layered on top of the independent heuristic approach. On the other hand,

our approach includes uncertainty quantification as an intrinsic aspect of the DCTO framework.

The third camp is the diverse collection of response surface methodologies (RSMs; Dean et al., 2017) used for optimization. RSMs operate by fitting a predictive model to an existing set of model runs, to form a computationally inexpensive meta-model which is then used to explore the model output. The concept of calibration to target outcomes that is built into DCTO is an example of an RSM, using GPs for its meta-model fit. Other popular versions of RSMs include efficient global optimization (EGO; Jones et al., 1998; Brochu et al., 2010) and stepwise uncertainty reduction (SUR; Geman and Jedynak, 1996; Villemonteix et al., 2009; Chevalier et al., 2014; Picheny, 2015; Miguel Hernández-Lobato et al., 2016; Picheny et al., 2019; Binois et al., 2019). EGO and SUR are both designed to facilitate sequential sampling from the system of interest in a search for the global optimum. They differ in their *acquisition functions*, which determine the location of the next sampling location throughout the optimization process. EGO finds the spot that maximizes the expected improvement (Mockus et al., 1978; Jones et al., 1998), whereas SUR’s acquisition function seeks to reduce the volume of excursion sets below the current best known solutions (Chevalier et al., 2014). Because they rely on sequential sampling, EGO and SUR are of limited utility when one is constrained to rely on a pre-existing set of observations, or in general when the observation locations cannot be chosen purely to suit the goal of optimization. Furthermore, the acquisition functions employed by

EGO and SUR attempt to balance exploitation (proposing a new sample location that optimizes system output) with exploration (proposing a location that promotes learning for subsequent rounds of sampling). As a result, although these acquisition functions constitute distributions of sampling locations, by their nature they are not interpretable as distributions of the *optimal* design settings for a given problem, and hence these distributions do not quantify uncertainty regarding the location of that optimum. By contrast, our approach (understood as a pure-exploitation method) quantifies remaining uncertainty regarding the location of the system optimum.

An example of an RSM more closely resembling our approach to design is described by Olalotiti-Lawal and Datta-Gupta (2018). Their approach defines a distribution which is designed to lie both on and near the Pareto front (PF) of the objective function and generates a posterior distribution which includes quantified uncertainties via Markov chain Monte Carlo (MCMC; Gelfand and Smith, 1990). However, the posterior distribution in that work is designed by the authors and is not dictated by the model itself; as such, its interpretability is not entirely clear. By contrast, our approach provides a posterior distribution based on the likelihood of the optimal design settings given the (hypothetical) observation of target outcomes y_t , and thus the uncertainty quantified by design using the KOH framework is model-driven and interpretable as uncertainty regarding the optimal values for the design inputs and the resulting system output.

The rest of the paper is organized as follows. Section 2 describes the difficulties

involved in extending KOH into a framework that incorporates both design and calibration, illustrating this by considering the failings of a naïve method for combining the two procedures, followed by a description of the proposed DCTO framework for extending KOH. Section 3 considers how DCTO may be useful in the case where sequential sampling is possible. In particular, sequential sampling with DCTO is attractive when the calibration parameter is known or suspected to be functionally dependent upon the design settings. We showcase the application of DCTO with sequential sampling using a synthetic example, comparing its results to that of a more traditional approach of design following calibration. In Section 4 we apply DCTO to a dynamic vibration system, using a set of experimental observations simultaneously to calibrate a finite element model and to select gain factor settings to achieve the optimal vibration isolation outcome, while demonstrating DCTO’s thorough quantification of the relevant uncertainties. Section 5 concludes with discussion of the results and thoughts about future directions.*

*Note that KOH is usually conceived of as a means of calibrating a computer model with respect to a set of experimental observations. However, the KOH framework, and by extension DCTO, are applicable more generally whenever one has access to both low-fidelity and high-fidelity sources of information and seeks to calibrate the former with respect to the latter. This includes the case in which both the high-fidelity and low-fidelity sources of information are computer models (e.g. with different levels of computational expense). For ease of exposition, we follow the common convention, and present DCTO in terms of calibrating a computer model using experimental observations. Nonetheless, in some cases (such as in our discussion of sequential sampling in Section 3), the methods discussed may apply more naturally in the context of employing two computer models of varying fidelity.

2 Dual calibration to target outcomes

The version of KOH considered here is that which finds a posterior distribution on a parameter of interest for calibration, $\boldsymbol{\theta}$, using a GP emulator with hyperparameters $\boldsymbol{\phi}_\eta$. Similarly, one may also use a GP prior with hyperparameters $\boldsymbol{\phi}_\delta$ to model discrepancy between the computer model $\eta(\cdot)$ and the true function $f(\cdot)$ that it represents. In the work described here, we employ stationary GPs with a Gaussian kernel covariance structure $C(\mathbf{x}, \mathbf{x}') = 1/\lambda \times \exp(-\beta(\mathbf{x} - \mathbf{x}')^2)$, so that $\boldsymbol{\phi}_\eta = [\beta, \lambda]$. In our adaptation, $\boldsymbol{\theta} = (\boldsymbol{\theta}_c, \boldsymbol{\theta}_d)$ is partitioned into parameters $\boldsymbol{\theta}_c$ to be calibrated and inputs $\boldsymbol{\theta}_d$ to be optimized for design purposes. Setting priors on $\boldsymbol{\theta}$ and on $\boldsymbol{\phi}_\delta$, we train the GP emulator on observations $\boldsymbol{\eta}$ and use MCMC to explore the distribution

$$\pi(\boldsymbol{\theta}, \boldsymbol{\phi}_\eta, \boldsymbol{\phi}_\delta | \mathcal{D}) \propto \pi(\mathcal{D} | \boldsymbol{\theta}, \boldsymbol{\phi}_\eta, \boldsymbol{\phi}_\delta) \times \pi(\boldsymbol{\theta}) \times \pi(\boldsymbol{\phi}_\eta) \times \pi(\boldsymbol{\phi}_\delta) \quad (1)$$

where $\mathcal{D} = (\boldsymbol{\eta}^T, \mathbf{y}^T)^T$ for some observations \mathbf{y} .

In a computer calibration problem, \mathbf{y} is a set of observations of the system modeled by $\eta()$. When calibrating to target outcomes as in DCTO, by contrast, y is a set of target outcomes representing the way that one wishes to induce the system to behave (rather than observations one has made of the system in reality). When one wishes to perform design leveraging a simulation model that also requires traditional calibration, then, one might consider combining the two approaches by using Equation (1) with $\mathbf{y} = (\mathbf{y}_r^T, \mathbf{y}_t^T)^T$, an array containing both real observations \mathbf{y}_r (for calibration)

and target outcomes \mathbf{y}_t (for design). However, this approach will not work, both because the inputs to be calibrated are typically not the same as the design settings under researcher control, and also because for successful calibration one must train one's model on observations of reality rather than on unobserved target outcomes.

Hence, model calibration and system design must be separated. An obvious choice here is to perform KOH calibration first, without involving any target outcomes, and then to use the calibrated model for model-assisted design. Under this approach, with observations \mathbf{y}_r of the system of interest, one would employ the model described in Equation (1) with $\boldsymbol{\theta} = \boldsymbol{\theta}_c$ (the parameters to be calibrated) and with $\mathcal{D} = \mathcal{D}_c = (\boldsymbol{\eta}^T, \mathbf{y}_r^T)^T$. The result would be a posterior distribution of $\boldsymbol{\theta}_c$ and of $\delta(\cdot)$, the systematic discrepancy between the computer model $\eta(\cdot, \cdot)$ and the true system $f(\cdot)$. These can be used to produce estimates $\widehat{\boldsymbol{\theta}}_c$ and $\widehat{\delta}(\cdot)$ such that $f(\mathbf{z}) \approx \eta(\mathbf{z}, \widehat{\boldsymbol{\theta}}_c) + \widehat{\delta}(\mathbf{z})$ for all \mathbf{z} in the domain of f . The result is a calibrated model $\eta_c(\mathbf{z}) = \eta(\mathbf{z}, \widehat{\boldsymbol{\theta}}_c) + \widehat{\delta}(\mathbf{z})$ which can be used for design.

With η_c in hand, one can partition \mathbf{z} into $(\mathbf{x}, \boldsymbol{\theta}_d)$ where $\boldsymbol{\theta}_d$ is the set of inputs over which one wishes to optimize, and \mathbf{x} are all other inputs in the operational domain, within which the calibrated model's predictions are reliable. We can write $\eta_c(\mathbf{z})$ as $\eta_c(\mathbf{x}, \boldsymbol{\theta}_d)$. Then one can perform design again using Equation (1), this time with $\boldsymbol{\theta} = \boldsymbol{\theta}_d$ and $\mathcal{D} = \mathcal{D}_t = (\boldsymbol{\eta}_c^T, \mathbf{y}_t^T)^T$ where $\boldsymbol{\eta}_c = \boldsymbol{\eta} + \widehat{\boldsymbol{\delta}} = \boldsymbol{\eta} + (\widehat{\delta}(\mathbf{z}_1), \dots, \widehat{\delta}(\mathbf{z}_n))^T$. Notice that a single set of simulator runs $\boldsymbol{\eta}$ can be used both for KOH and for subsequent CTO. A crucial difference between calibration and design is that for the design step

one would not attempt to model any systematic discrepancy between η_c and f , since an estimate of that discrepancy is already included in η_c . For the purposes of Equation (1), this amounts to setting a degenerate prior on ϕ_δ at 0.

A problem with the above-described approach of performing calibration prior to a separate design optimization is that relying on static calibration estimates $\widehat{\boldsymbol{\theta}}_c$ ignores uncertainty remaining after calibration with respect to the true value of $\boldsymbol{\theta}_c$. In order to produce results that take into account all sources of uncertainty, it is necessary to integrate calibration and design, so that the uncertainty remaining from calibration is propagated through the design process. This can be accomplished either asynchronously (so that the posterior distribution of $\widehat{\boldsymbol{\theta}}_c$ is sampled while undertaking design) or, for lower computational overhead, synchronously (so that a single MCMC run is used to perform both calibration and design). In either case, it will be useful to produce an integrated model which describes the use of both procedures, and which makes clear the relationship between them. This integrated model will also serve to demonstrate the unified framework underlying the synchronous approach.

For this purpose, consider η as having three inputs $(\mathbf{x}, \mathbf{t}_c, \mathbf{t}_d)$ where \mathbf{t}_c denotes the parameters targeted for KOH calibration, \mathbf{t}_d denotes the input settings targeted for design, and \mathbf{x} denotes the remaining controllable inputs. If η can be run quickly, then we use it directly in MCMC. However, if it is computationally expensive, we employ a surrogate by setting a Gaussian process (GP) prior on η with mean $m_\eta(\mathbf{x}, \mathbf{t}_c, \mathbf{t}_d)$ and covariance function $C_\eta((\mathbf{x}, \mathbf{t}_c, \mathbf{t}_d), (\mathbf{x}', \mathbf{t}'_c, \mathbf{t}'_d))$. From here on in this discussion,

assume that a GP surrogate is used for η . We model the systematic discrepancy between η and f at the true value of $\mathbf{t}_c = \boldsymbol{\theta}_c$ with another GP prior $\delta(\cdot, \cdot)$ having mean $m_\delta(\mathbf{x}, \mathbf{t}_d)$ and covariance function $C_\delta((\mathbf{x}, \mathbf{t}_d), (\mathbf{x}', \mathbf{t}'_d))$. In addition to systematic discrepancy between η and reality, measurement error ϵ_r may be included in the model for real observations \mathbf{y}_r , and additional Gaussian observation error ϵ_d may be included for target outcomes \mathbf{y}_t .

The purpose of additional observation error ϵ_d is twofold. Depending on the distribution of ϵ_c , the target outcomes \mathbf{y}_t may or may not be possible outputs of a model that lacks ϵ_d . Including ϵ_d ensures that there is nonzero probability of an observation falling in the vicinity of the targets. Secondly, including ϵ_d and estimating its variance σ_d^2 provides computational benefits. For example, even if the target outcomes are compatible with a model that does not include ϵ_d , they may (depending on the choice of targets) be extreme outliers to the extent that the relevant likelihoods are small enough to generate significant numerical errors during MCMC. In terms of the interpretation of the model, adding ϵ_d amounts to supposing that the counterfactual target outcomes were observed with greater than usual observation error, where that additional error is distributed as $N(0, \sigma_d^2)$. Though it is not necessary to assume that ϵ_c is Gaussian, for simplicity of presentation we assume here that it is distributed as $N(0, \sigma_c^2)$. Finally, we assume that η, δ, ϵ_c and ϵ_d are all mutually independent.

A collection of simulation runs is needed to train the GP code surrogate. Let $(\mathbf{x}_s, \mathbf{t}_{cs}, \mathbf{t}_{ds})$ be the design matrix for the settings of the simulation runs, and let \mathbf{y}_s

denote the output of these runs. Similarly, let \mathbf{y}_r be observations made at $\mathbf{x}_r, \mathbf{t}_{dr}$, and let \mathbf{y}_t be target outcomes we wish to observe at \mathbf{x}_t . Finally, let $\mathbf{y} = (\mathbf{y}_s^T, \mathbf{y}_r^T, \mathbf{y}_t^T)^T$, and $\mathbf{1}$ a vector of ones. Then it follows that $\mathbf{y} \sim N(\mathbf{m}, \mathbf{C})$, where

$$\mathbf{m} = \begin{pmatrix} m_s(\mathbf{x}_s, \mathbf{t}_{cs}, \mathbf{t}_{ds}) \\ m_s(\mathbf{x}_r, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t}_{dr}) + m_\delta(\mathbf{x}_r, \mathbf{t}_{dr}) \\ m_s(\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T) + m_\delta(\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_d^T) \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23} \\ \mathbf{C}_{31} & \mathbf{C}_{32} & \mathbf{C}_{33} \end{pmatrix},$$

$$\mathbf{C}_{11} = C_\eta((\mathbf{x}_s, \mathbf{t}_{cs}, \mathbf{t}_{ds}), (\mathbf{x}_s, \mathbf{t}_{cs}, \mathbf{t}_{ds}))$$

$$\mathbf{C}_{21} = C_\eta((\mathbf{x}_s, \mathbf{t}_{cs}, \mathbf{t}_{ds}), (\mathbf{x}_r, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t}_{dr}))$$

$$\mathbf{C}_{31} = C_\eta((\mathbf{x}_s, \mathbf{t}_{cs}, \mathbf{t}_{ds}), (\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T))$$

$$\mathbf{C}_{12} = \mathbf{C}_{21}^T$$

$$\mathbf{C}_{22} = C_\eta((\mathbf{x}_r, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t}_{dr}), (\mathbf{x}_r, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t}_{dr})) + C_\delta((\mathbf{x}_r, \mathbf{t}_{dr}), (\mathbf{x}_r, \mathbf{t}_{dr})) + \sigma_c^2 \mathbf{I}$$

$$\mathbf{C}_{32} = C_\eta((\mathbf{x}_r, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t}_{dr}), (\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T)) + C_\delta((\mathbf{x}_r, \mathbf{t}_{dr}), (\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_d^T))$$

$$\mathbf{C}_{13} = \mathbf{C}_{31}^T$$

$$\mathbf{C}_{23} = \mathbf{C}_{32}^T$$

$$\mathbf{C}_{33} = C_\eta((\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T), (\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T)) + C_\delta((\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_d^T), (\mathbf{x}_t, \mathbf{1}\boldsymbol{\theta}_d^T)) + \sigma_c^2 \mathbf{I} + \sigma_d^2 \mathbf{I}$$

Note that when \mathbf{y}_t and \mathbf{x}_t are empty and \mathbf{m}, \mathbf{C} reduce respectively to their first two and upper two-by-two block elements, this is simply the KOH framework. Thus, DCTO is an extension of the KOH framework to include design using target outcomes.

A primary benefit of DCTO is that the design process includes quantification of all sources of uncertainty. Performing calibration and then subsequently undertaking design using static estimates for $\hat{\boldsymbol{\theta}}_c$ and $\hat{\delta}$ does not properly account for the uncertainty surrounding the estimates. Another benefit of the combined approach appears in cases in which the model is misspecified in failing to account for functional dependence of $\boldsymbol{\theta}_c$ on $\boldsymbol{\theta}_d$. In such cases, one may be interested only or primarily in the value of

$\boldsymbol{\theta}_c$ at the optimal value of $\boldsymbol{\theta}_d$. If one has the freedom to sample adaptively from the true system, then this freedom can be applied in DCTO to concentrate samples disproportionately in the region of interest. This idea is explored further in Section 3.

For DCTO, we employ modularity in the manner of Liu et al. (2009). A modular analysis intentionally falls short of being a full Bayesian analysis, either for computational benefits, or to quarantine “suspect” aspects of the model, so that the posterior distributions of parameters of interest are robust to model misspecification. The target outcomes \mathbf{y}_t are precisely such a suspect source of Bayesian learning—they are by their nature extreme outliers, and hence are a poor guide both for estimating the hyperparameters of the GP emulator and for estimating the parameter $\boldsymbol{\theta}_c$. To modularize DCTO, we estimate the emulator hyperparameters via maximum likelihood, and we refrain from including \mathbf{y}_t in the updates of $\boldsymbol{\theta}_c$ during MCMC. That is, rather than calculating the likelihood of a proposed sample $t_c^{(i+1)}$ at step i of the MCMC using $\mathbf{y} = (\mathbf{y}_s^T, \mathbf{y}_r^T, \mathbf{y}_t^T)^T$, we instead calculate its likelihood using only $\mathbf{y} = (\mathbf{y}_s^T, \mathbf{y}_r^T) \sim N(\mathbf{m}_r, \mathbf{C}_r)$, where \mathbf{m}_r and \mathbf{C}_r are respectively the upper two and upper-left two-by-two components of \mathbf{m} and \mathbf{C} . Such modularization ensures that all Bayesian learning of $\boldsymbol{\theta}_c$ is based upon the real observations rather than upon \mathbf{y}_t .

3 Dependence of θ_c on θ_d

In many cases of computer model calibration, it is known or suspected that the value of one or more calibration parameters are functionally dependent upon the values of other model inputs (Atamturktur and Brown, 2015; Atamturktur et al., 2017; Ezzat et al., 2018). If one is interested to understand the functional form of the calibration parameter, then state-aware methods can be used to arrive at such an estimate (Atamturktur and Brown, 2015; Atamturktur et al., 2017; Brown and Atamturktur, 2018).

In a case where the calibration parameter is functionally dependent upon the design settings, one might be interested only to know the value of the calibration parameter in the optimal design region. When calibration and design are undertaken simultaneously, as in DCTO, the machinery of state-aware calibration is not needed, and effort is better spent focusing on estimating the fixed calibration parameter value in the region of interest. In such a case, it is preferable that one's calibration be founded on observations for which the design settings are in the optimal design region. This will allow one to calibrate the model using observations taken from the region of design interest, so that the calibration takes on values that are most applicable in that region.

When observations may be made adaptively, other RSM approaches such as EGO (Jones et al., 1998; Brochu et al., 2010) or SUR (Geman and Jedynak, 1996; Villemonteix et al., 2009; Chevalier et al., 2014; Picheny, 2015; Miguel Hernández-Lobato

et al., 2016; Picheny et al., 2019; Binois et al., 2019) may be more efficient than the KOH framework for estimating optimal design settings, though the KOH framework offers more interpretable and model-driven uncertainty quantification. Further, RSM approaches in general do not include tools to accommodate the case in which a model stands in need of calibration as well as optimization. DCTO provides such a framework for combined calibration and design.

Therefore, we now consider under the lens of DCTO the case in which the design settings of the observations of the true system may be chosen adaptively. The use of DCTO with adaptive sampling is potentially of greatest use when it is known or suspected that the calibration parameter is a function $\theta_c(t_d)$ of the design setting t_d , and particularly when interest focuses on learning the optimal design setting θ_d and the corresponding value $\theta_c(\theta_d)$ of the calibration parameter. The process of performing DCTO with adaptive sampling is described in Algorithm 1. When adaptively

Algorithm 1: DCTO with adaptive sampling

- 1 Set $\mathbf{y} = [\mathbf{y}_r^T \mathbf{y}_t^T]^T$ where \mathbf{y}_t are the target outcomes and $\mathbf{y}_r = []$ is an empty array.
- 2 Begin MCMC burn-in. Set $i = 1$. Let m be the budget of function evaluations. While $i \leq m$:
 - 2.1 Complete n iterations of MCMC burn-in (where e.g. $n = 100$).
 - 2.2 Draw $\hat{\theta}_d$ from the available size $n \cdot i$ sample of $t_d | \mathbf{y}$.
 - 2.3 Evaluate $f(\mathbf{x}_i, \hat{\theta}_d)$.
 - 2.4 Set $\mathbf{y}_r = [\mathbf{y}_r^T f(\mathbf{x}_i, \hat{\theta}_d)]^T$.
- 3 Continue burn-in until convergence.
- 4 Draw a sample of desired size from the posterior distributions of θ_c, θ_d .

evaluating the objective function, the locations of the input settings \mathbf{x}_i which are not being optimized for design can be selected to maximize distance from previous

observations, or these locations can be predetermined according to a space-filling design over the domain of non-design inputs. The result of applying this algorithm is that observations are concentrated around the design settings of interest, so that the unknown calibration parameter values in those observations are concentrated around the value $\theta_c(\theta_d)$.

To demonstrate the use of DCTO with adaptive sampling in a case of functional dependence of the calibration parameter on design settings, we use the function of three inputs

$$f_0(x, t_c, t_d) = x / (t_d^{t_c - 1} \exp(-0.75t_d) + 1). \quad (2)$$

Figure 1 shows the output of this function for $x = 1$ over the range $(t_c, t_d) \in [1.5, 4.5] \times [0, 5]$. For any value of x and t_c , the optimal (minimizing) value of t_d is $(4/3)(t_c - 1)$. Suppose that the calibration parameter’s “true” value is functionally dependent on the design input, with the relationship:

$$\theta_c(t_d) = 2.25 - .75 \frac{\exp\left(40\left(\frac{t_d-1.5}{.75} - .5\right)\right)}{1 + \exp\left(40\left(\frac{t_d-1.5}{.75} - .5\right)\right)} \quad (3)$$

which would be unknown in a real application. Figure 2 shows this relationship. Figure 3 shows the locations of the true and optimal values (respectively) of θ_c and θ_d . There it is clear that the true value of θ_c is far from optimal, in the sense that if this value *were* within our control (which, being a calibration parameter, it is not),

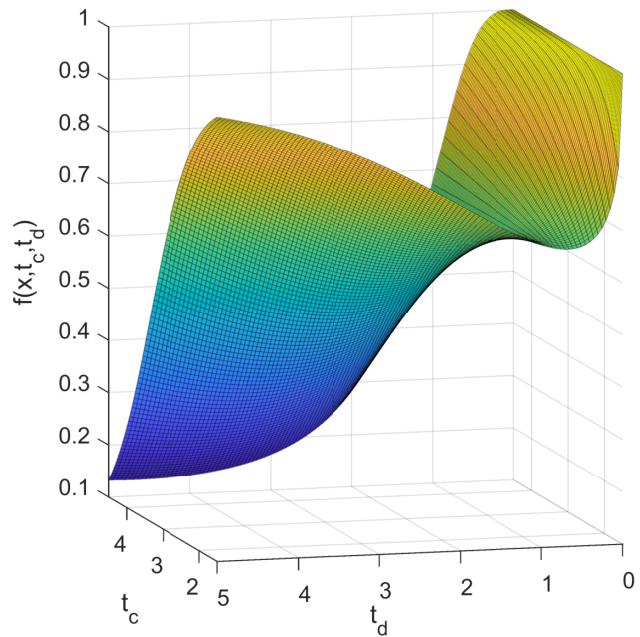


Figure 1: Example computer model output over the support of the calibration parameter t_c and the design parameter t_d .

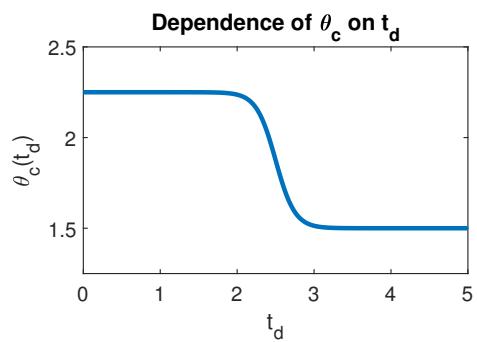


Figure 2: True value of the calibration parameter θ_c for each value in the domain of t_d

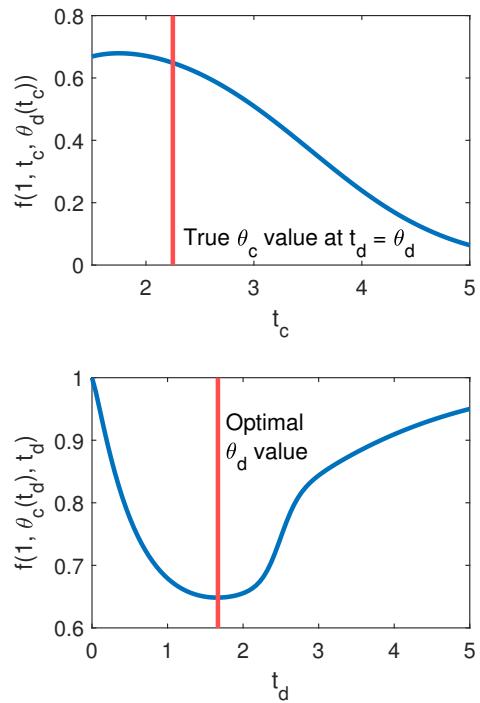


Figure 3: The top plot shows the computer model output at $x = 1$ and optimal design setting for each value of the calibration parameter t_c . The bottom plot show the model output at $x = 1, t_c = \theta_c(t_d)$ for each value of the design parameter t_d .

we would prefer to place it at the upper end of its support, at 4.5. Thus η showcases the ability of DCTO to perform simultaneously both calibration and design in the case when our “truth-seeking” goals and our design goals are in tension.

We apply DCTO to four versions of the problem. First, we assume that η is free from discrepancy; i.e. that $\eta(x, \theta_c, t_d)$ is an unbiased estimator of the “true” system $f(x, t_d)$. The other three versions each assume that η suffers from some form of discrepancy. Let f_1, f_2, f_3 denote the “true” systems in these three cases. We set

$$f_1(x, t_d) = \eta(x, \theta_c, t_d) (1 - a(x - .5)(x - 1)/x)) \quad (4)$$

$$f_2(x, t_d) = \eta(x, \theta_c, t_d) - a(x - .5)(x - 1) \left(t_d - \frac{4}{3} \right)^2 + b \quad (5)$$

$$f_3(x, t_d) = \eta(x, \theta_c, t_d) + axt_d + b \quad (6)$$

where a, b are constants which determine how severe the discrepancy is in each case. The function f_1 has a multiplicative discrepancy dependent only on x and a . This discrepancy does not affect the optimal value of t_d . The discrepancies of f_2 and f_3 are both additive. Figure 4 shows the discrepancies for two different versions (corresponding to different settings of (a, b)) of each f_i .

We apply DCTO with and without adaptive sampling to each of seven cases, without using an emulator: the non-discrepancy case, and the two different versions of each f_i shown in Figure 4. In each case, we gather 30 “observations” of f_i on a latin hypercube design over the supports of x and t_d , setting θ_c equal to its “true” value of

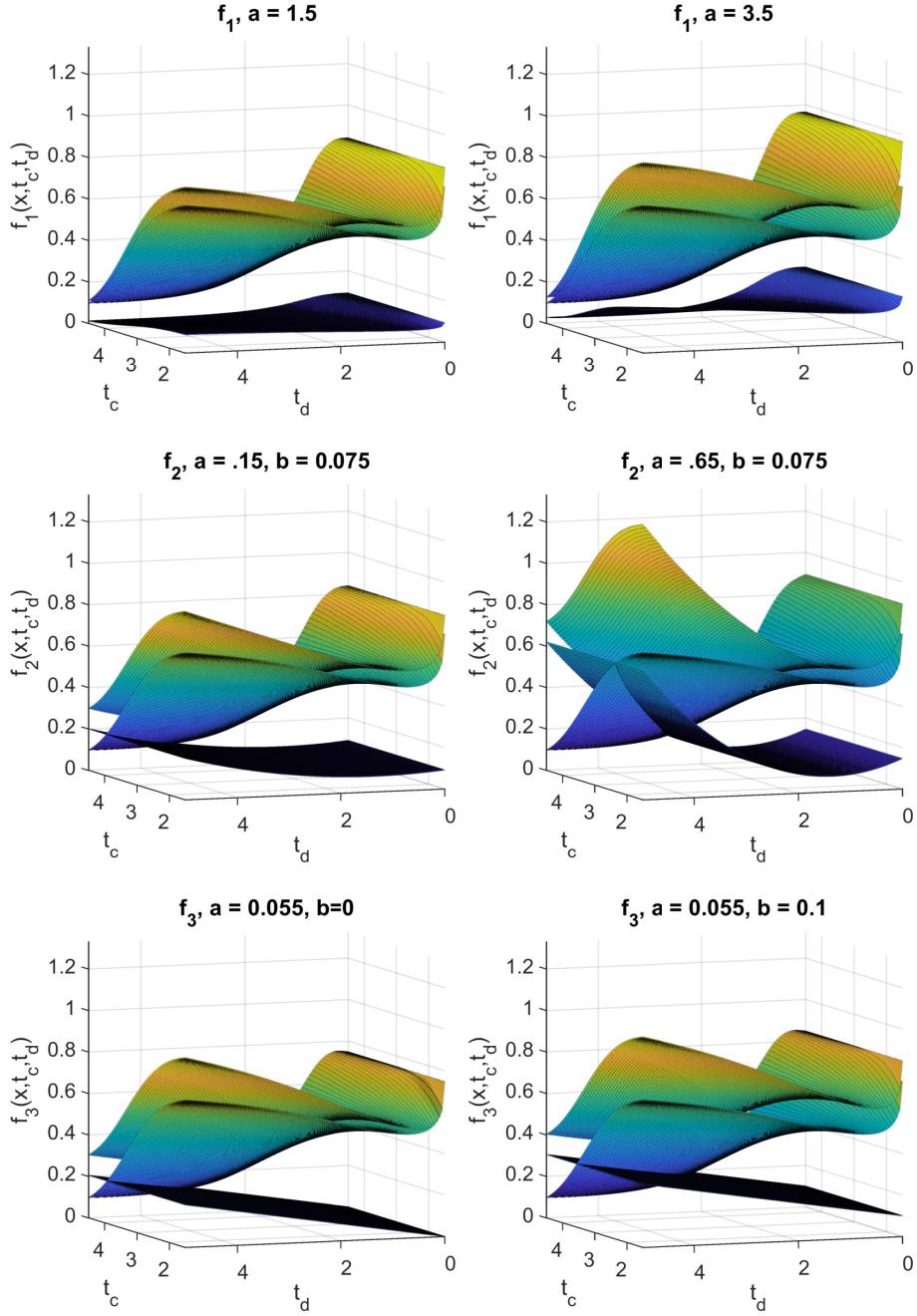


Figure 4: The i^{th} row shows f_i (the objective function with discrepancy), η (the computer model), and the discrepancy $f_i - \eta$, all at $x = 0.75$. In each row, a less aggressive version of the discrepancy appears on the left, and a more aggressive on the right. In each plot, the topmost surface is f_i , the middle surface is η , and the bottom surface is the discrepancy $f_i - \eta$.

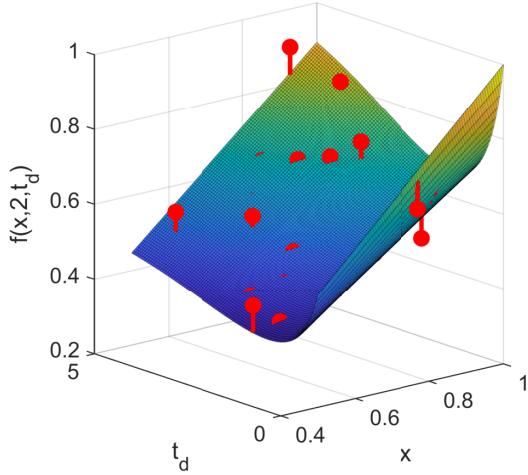


Figure 5: Noisy observations of the system, and the true system mean, for $f = f_0$ (no discrepancy).

$\theta_c(t_d)$. After standardizing the response to have mean 0 and standard deviation 1, we add i.i.d. $N(0,0.05)$ noise to the response. An example of the resulting “observations” from non-adaptive DCTO, with noise, appears in Figure 5. We carry out DCTO using Metropolis-Hastings-within-Gibbs MCMC, drawing 8000 realizations each (discarding the first 4000 as burn-in) of $t_c, t_d, \boldsymbol{\rho}_\delta, \lambda_\delta, \sigma_d^2$, where $\boldsymbol{\phi}_\delta = (\boldsymbol{\rho}_\delta^T, \lambda_\delta)^T$. For the adaptive sampling application of DCTO, we begin the MCMC with 0 observations of f_i , making a new observation after every 100 steps of MCMC until we reached the total budget of 20. An example of the resulting difference between the adaptive sampling approach and relying on a space filling design, with regard to the sampling distribution of our observations of the objective function, appears in Figure 6. There, one can see that the adaptive sampling approach manages to expend its budget on observations that are near to the region of design interest. This explains the superior performance of adaptive sampling (discussed below) in both design and in calibration (since the

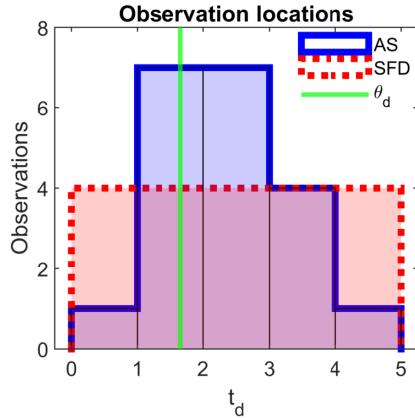


Figure 6: Design input values for observations made under the adaptive sampling approach (AS) and under a space-filling design (SFD), along with the optimal value θ_d of the design input.

value of the calibration parameter is dependent upon that of the design input). This ameliorative effect would likely be even greater in a higher-dimensional case, in which a space-filling design would (due to the curse of dimensionality) tend to generate observations even farther from the region of design interest.

In both versions of DCTO, we modularize the analysis by drawing each of $\boldsymbol{\theta}_c$, $\boldsymbol{\rho}_\delta$, λ_δ using the likelihood based only on $(\mathbf{y}_s^T, \mathbf{y}_r^T)^T$ rather than on $(\mathbf{y}_s^T, \mathbf{y}_r^T, \mathbf{y}_t^T)^T$. Convergence was verified visually and by the Gelman-Rubin statistic (≈ 1.01 ; Gelman and Rubin, 1992).

The resulting optimal design settings and calibration parameter value at the optimum vary in the discrepancy cases, though $\theta_c(\theta_d)$ is near 2.16 in each case. Representative results from performing DCTO with adaptive sampling in each discrepancy case appear in Figure 7, along with results from applying DCTO non-adaptively (using a space-filling set of observations). A summary of the results of thirty applications

Table 1: Posterior root mean square error (RMSE) for the calibration variable θ_c and the design variable θ_d , for DCTO with adaptive sampling (AS) and a predetermined space-filling design (SFD). The estimator $\hat{\theta}_i$ is the posterior mean of t_i for $i = c, d$. For each f_i , a and b control the size of the discrepancy as specified in Equations (4,5,6).

	$\hat{\theta}_c$ RMSE		$\hat{\theta}_d$ RMSE	
	AS	SFD	AS	SFD
f_0 (no discrepancy)	0.188	0.433	0.163	0.479
$f_1, a = 1.5$	0.233	0.32	0.243	0.414
$f_1, a = 3.5$	0.188	0.247	0.213	0.393
$f_2, a = .15, b = .075$	0.221	0.263	0.187	0.348
$f_2, a = .65, b = .075$	0.228	0.16	0.183	0.206
$f_3, a = .055, b = 0$	0.452	0.506	0.182	0.329
$f_3, a = .055, b = .1$	0.448	0.468	0.167	0.292

of DCTO both with and without adaptive sampling, for each of the discrepancy cases,

appears in Table 1.

The results show superior performance for the adaptive sampling DCTO over DCTO using a space-filling design of experiments for the true phenomenon (or high-fidelity model, in a case of calibrating a low-fidelity model to use for design purposes). The adaptive DCTO posterior means have lower RMSEs in all cases for θ_d , and in all cases except one for θ_c . This demonstrates a useful robustness of adaptive DCTO to model misspecification, specifically in the case that the model treats as constant a calibration parameter that is more properly understood as functionally dependent upon other model inputs. By using the CTO-driven estimate $\hat{\theta}_d$ to sample from the region of interest, DCTO learns from observations such that $\theta_c(\hat{\theta}_d)$ is near to the value $\theta_c(\theta_d)$. This promotes better calibration with respect to the region of interest,

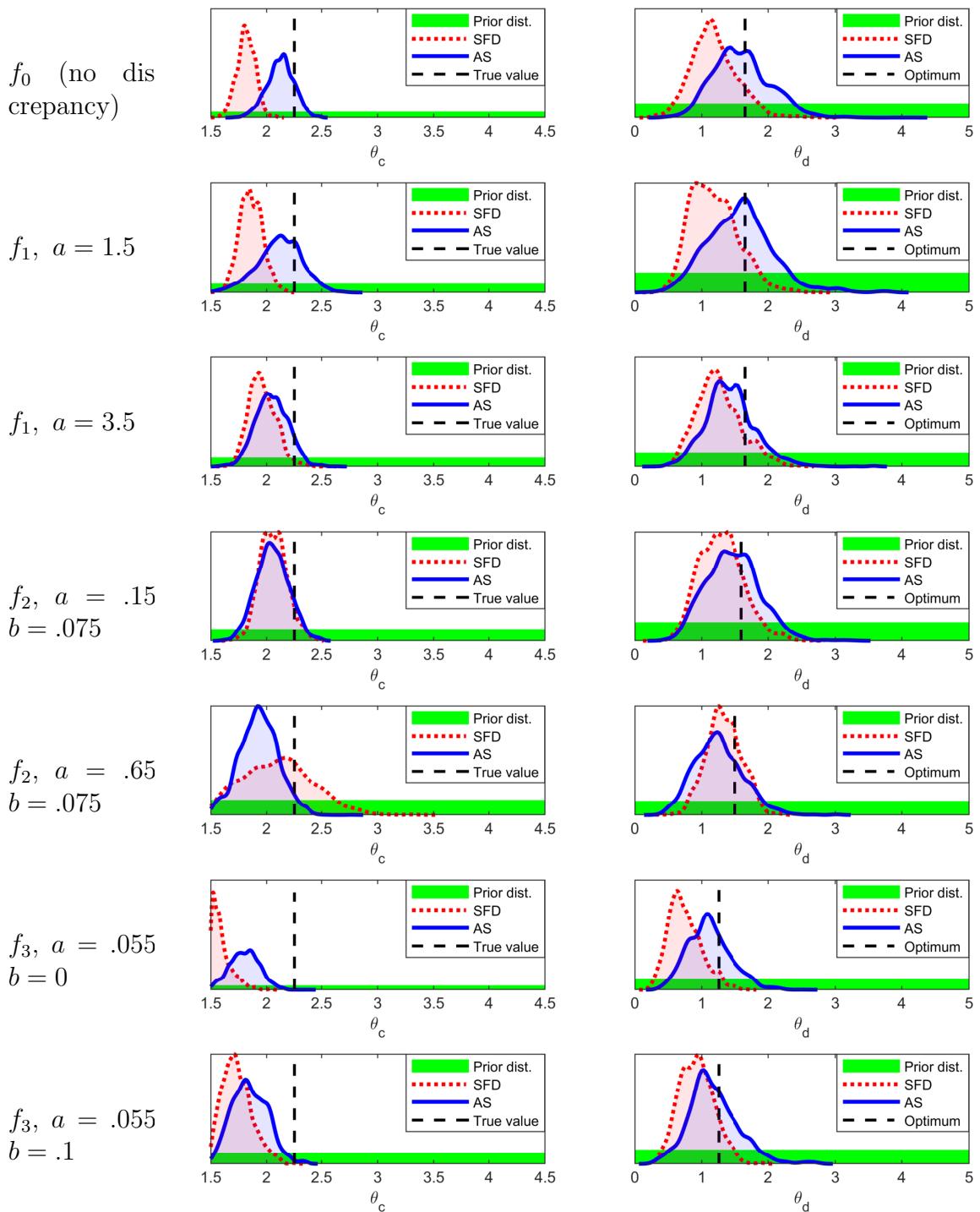


Figure 7: Prior and posterior distributions of the calibration parameter θ_c and design parameter θ_d , along with their true/optimal values, for DCTO with adaptive sampling (AS) and with predetermined space-filling design (SFD) in each of the cases studied. For each f_i , a and b control the size of the discrepancy as specified in Equations (4,5,6).

and thereby better estimation of the optimal design settings. By relying on DCTO rather than on performing KOH using samples gathered using heuristic optimization methods, or other RSM approaches, we achieve these estimates with quantification of all relevant model-driven uncertainty with respect to the values of θ_c and θ_d .

4 Case study application: vibration isolation design

The application of DCTO methodology is demonstrated on a vibration isolation design problem. Vibration isolation relies on the balance of inertia, damping, and stiffness properties where, in active vibration isolation, an additional active gain factor enhances the system's damping behavior. To achieve the optimal vibration isolation outcome, the design engineer typically specifies the resonance and isolation frequencies and then balances mass, damping, stiffness, and the gain factor.

4.1 Case study problem

The experimental dynamical system studied herein is a one-mass oscillator subjected to passive and active vibration isolation (Figure 8). The system consists of a rigid rectangle frame, a rigid mass held by four identical orthogonally placed leaf springs mounted to the frame, and a voice coil actuator (VCA) for passive and active damping.

In Figure 9a and as a simplified model, a rigid mass m oscillates in the z -direction

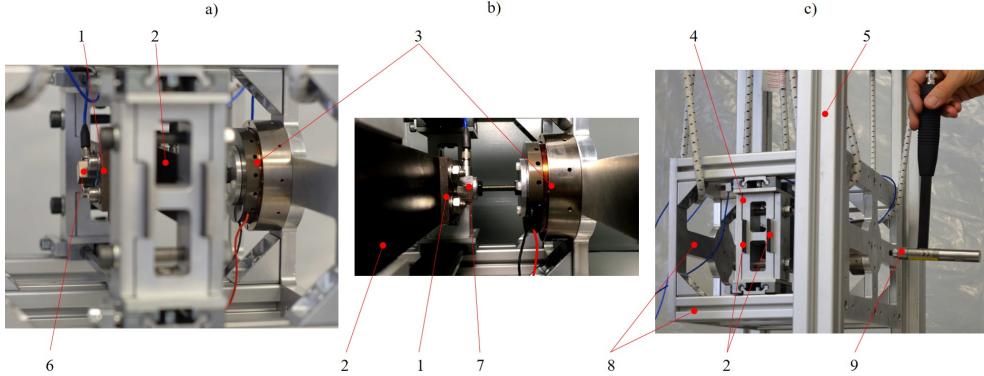


Figure 8: The physical test rig for the dynamic vibration system. The test setup is shown from three views. Depicted here are the rigid mass (1), one leaf spring (2), VCA (3), fixed leaf spring support (4), mount (5) to suspend the frame (8), acceleration sensor $S_{a;z}$ (6), force sensor $S_{F_{VCA}}$ (7), rigid frame (8), and a modal hammer (9) with a force sensor S_F to excite the frame. The sensor $S_{a;w}$ to measure the acceleration of the frame is on the inner side of the frame near the location where the impulse hammer hits ($S_{a;w}$ not visible in the figure).

due to a base point excitation $w(t)$. A damper with the damping coefficient b and a spring element with a stiffness constant k connect the mass to the base point. The damper and spring provide the system's internal passive damping force, active damping force, and stiffness force $F_b = b[\dot{z}(t) - \dot{w}(t)]$, $F_a = -g\dot{z}(t)$, $F_k = k[z(t) - w(t)]$ with F_a derived from a simple velocity feedback control with the gain factor g .

The inhomogeneous differential equation of the one-mass oscillator's motion in Figure 9a can be written as

$$\begin{aligned} \ddot{z}(t) + \left[2D_p\omega_0 + \frac{g}{m} \right] \dot{z}(t) + \omega_0^2 z(t) &= 2D_p\omega_0 \dot{w}(t) + \omega_0^2 w(t) \\ &= \omega_0^2 r(t) \end{aligned} \quad (7)$$

using the abbreviation $2D_p\omega_0 = \frac{b}{m}$, and $\omega_0^2 = \frac{k}{m}$ including the damping ratio D_p from

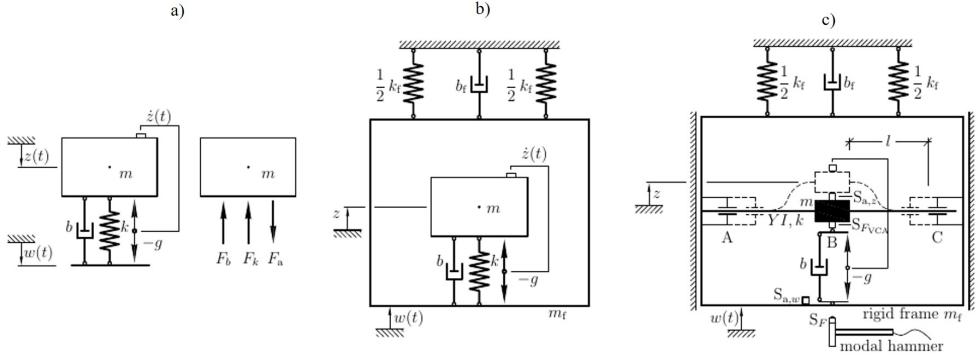


Figure 9: Schematic diagram of the test rig for the dynamic vibration system: (a) simplified schematic representation of the one-mass oscillator, (b) one-mass oscillator with an additional frame as the base point, (c) schematic representation of the real test setup.

passive damping, with $0 < D_p < 1$, and the angular eigenfrequency ω_0 . The term $\omega_0^2 r(t)$ in (7) is the excitation function, which, in this case, is the linear combination of the damper and spring base point excitation $2D_p\omega_0\dot{w}(t) + \omega_0^2 w(t)$.

Figure 9b depicts the laboratory set-up used in this study, in which a rigid frame with mass m_f serves as a base point structure. The frame is fixed by a gliding support assumed to have no friction perpendicular to the z -direction. The frame is constrained by a damper with the damping coefficient b_f and springs with a total stiffness k_f in the z -direction, and in the same plane.

In the laboratory application, the frame suspends from a rigid mount via elastic straps vertical to the z -direction, allowing the frame to move freely in the z -direction as shown in Figure 8c. The idealized damping b_f and k_f that constrain this movement are relatively small, compared to the b and k of the mass. The frame moves in a translational z -direction because of a time-dependent translational excitation

displacement $w(t)$ in the z -direction. As shown in Figure 9c, the frame retains two supports that fix a leaf spring at its ends at A and C, with the effective bending length l on sides A-B and B-C, with the rigid mass m in the center position at B. The leaf spring is the practical realization of the spring elements in Figure 9a and b. Its stiffness $k^* = 12EI/l^3$ is a function of the bending stiffness EI , where E is the Young's modulus of the leaf spring made from carbon fiber reinforced polymer (CFRP), I is the geometrical moment of inertia, and l is the length of the leaf spring. Two leaf springs are mounted in parallel with length l on each side of A-B and B-C (see Figures 8c and 9c). With four leaf springs, the total stiffness becomes $k = 4k^*$. The two supports at A and C in Figure 9c are adjustable along l to tune the leaf spring's bending deflection and therefore its effective stiffness k .

A VCA realizes an electromotive force F_{VCA} as the passive damping and the active force F_b and F_a (Figure 9c). The force sensor $S_{F_{\text{VCA}}}$ at B in Figure 9c measures the sum of forces F_b and F_a acting on the moving mass m . The acceleration sensors $S_{a;z}$ and $S_{a;w}$ measure directly the accelerations of mass and frame, \ddot{z} and \ddot{w} . The accelerations are transformed into velocities \dot{w} and \dot{z} by numerical integration in the Simulink-dSpace environment. The masses of $S_{a;z}$, $S_{F_{\text{VCA}}}$ and parts of the leaf spring are included in mass m (Table 2).

Figure 9c also shows a modal hammer with a force sensor S_F to excite the frame.

Table 2: Geometrical, mass, and material values of each component in the vibration isolation test rig

Category	Property		Variable Value	Unit
Rigid frame structure	sum mass	m_f	6.2073	kg
Vibrating rigid mass	sum mass, min	m	0.7853	kg
	20x add. weights, small	m_{ws}	0.0760	kg
	24x add. weights, large	m_{wl}	0.2880	kg
	sum mass, max	m	1.1493	kg
Geometry	leaf spring length, min	l	0.04	m
	leaf spring length, max	l	0.08	m
	leaf spring cross section, width	d	0.04	m
	leaf spring cross section, height	h	0.11	m
Material	Elastic modulus	E	$6.2 \cdot 10^9$	N/m ²
	stiffness CFRP, min	k	25,788.1	N/m
	stiffness CFRP, max	k	206,305.0	N/m
VCA	passive damping coefficient, min b		16	Ns/m
	passive damping coefficient, max b		130	Ns/m
	passive damping ratio, min	D_p	0.0481	-
	passive damping ratio, max	D_p	0.628	-
	active gain factor, min	g	0	Ns/m
	active gain factor, max	g	95	Ns/m

The hammer creates the impulse force

$$\hat{F}(t_0) = \int_{-\infty}^{\infty} F(t) \cdot \delta(t - t_0) dt, \quad (8)$$

including the Dirac-impulse function $\delta(t - t_0)$ that leads to the vibrational response of the frame

$$w(t) = \frac{\hat{F}(t_0)}{m_f \omega_{D,f}} \cdot e^{-D_f \omega_{0,f} t} \sin \omega_{D,f} t, \quad (9)$$

in the time domain, with damping ratio D_f , angular eigenfrequency $\omega_{0,f}$ and damped angular eigenfrequency $\omega_{0,f}$ of the frame's movement in z -direction. (9) is only valid for low damping $0 < D_f < 1$. This leads to the total vibration response $z(t) = r_0 \{1 - e^{-D_{\omega_0} t} [\cos \omega_D t - D \frac{\omega_0}{\omega_D} \sin \omega_D t]\}$.

The particular solution r_0 is part of the general excitation function $\omega_0^2 r(t)$ in (7), which takes the form of an excitation step function $r(t) = r_o \sigma(t - t_0)$ when multiplied with the unit step function $\sigma(t - t_0)$ as the integral of the Dirac-impulse function $\delta(t - t_0)$ in (8). From the relation $2D_p \omega_0 \dot{w}(t) + \omega_0^2 w(t) = \omega_0^2 r(t)$ in (7), it follows that $r_0 = \frac{1}{\omega_0} 2D_p \dot{w}_0 + w_0$ with the velocity \dot{w}_0 and displacement w_0 at $t = t_0$ that are derived from (9).

In this demonstration, the design problem is formulated with the gain factor g being the design parameter θ_d . The elastic modulus E of the four leaf spring is assumed to be poorly known and is assigned as the calibration parameter θ_c . The mass m of the system that needs to be vibration isolated is treated as the control

Table 3: A variety of experiment tests and 5-times averaged results

Case	Control Parameter x	Design Parameter θ_d	5-times Avg. System Response y
Variable unit	Mass (kg)	Gain factor (Ns/m)	Overall damping ratio [†] (-)
1	1.1493	0	0.0523
2	0.9653	0	0.0481
3	0.7853	0	0.0549
4	1.1493	8	0.0798
5	0.9653	8	0.0864
6	0.7853	8	0.0871
7	1.1493	41	0.308
8	0.9653	41	0.264
9	0.7853	41	0.259
10	1.1493	95	0.542
11	0.9653	95	0.527
12	0.7853	95	0.628

parameter x , and the damping ratio is the design objective y .

4.2 Experimental observations

For the dual model calibration, 12 operational conditions for the test rig are designed for varying values of the mass m and gain factor g (shown in Table 3). To excite the test rig, an impulse force is applied in the translational z -direction via a modal hammer. The time history response of the hammer excitation is shown in Figure 10. Figure 11 shows the acceleration response $\ddot{z}(t)$ of the mass, as measured by the acceleration sensor $S_{a,z}$. Since the rigid frame is constrained by a spring of small stiffness in the z -direction, the resulting relatively low resonance frequency of

[†]“Overall damping ratio” refers to the combination of active and passive damping.

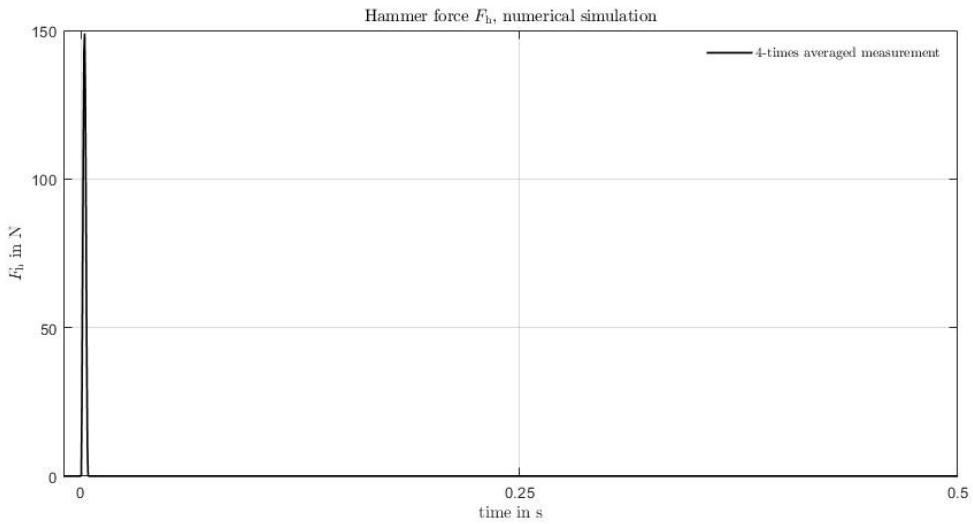


Figure 10: Schematic diagram of the applied impulse force in the time domain.

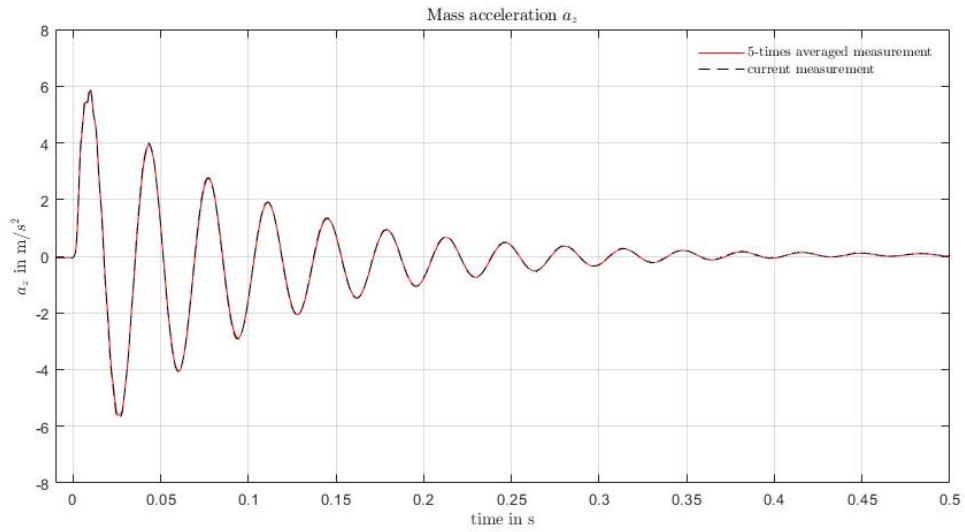


Figure 11: Five-times averaged acceleration response of the rigid mass in the time domain.

the frame (≈ 1.5 Hz) does not significantly affect the mass vibration with its higher eigenfrequency (> 20 Hz) when vibrating in the z -direction. The low frequency content is filtered out in the measurement chain. The hammer impact is repeated 5 times, and the impact force and the system response measurements are averaged.

One significant character of an oscillatory system is its damping (i.e. how rapidly a vibration system will decay after the initial excitation). The damping ratio, a dimensionless measure that describes the damping level, is calculated as $D = (1 + (2\pi/\delta)^2)^{-1/2}$, where $\delta = \frac{1}{n} \ln(\ddot{z}(t)/\ddot{z}(t + nT))$, $\ddot{z}(t)$ is the 1st peak value of mass acceleration, $\ddot{z}(t + nT)$ is the $n + 1^{\text{th}}$ peak value of mass acceleration, and n is the number of peak intervals. δ is the logarithmic decrement, which is used to compute the damping ratio D_p . By following these two equations, the system responses under various numerical simulations are summarized in Table 3.

4.3 Numerical investigation

To fully explore the domain of the control parameter in this dual model calibration problem, a finite element model of the one-mass oscillator is built in ANSYS v. 2018 (Figure 12). The frame and oscillatory mass are represented by a linear solid element type C3D8R in ABAQUS. Both the frame and the mass are assigned very high stiffness values to reflect rigid body behavior. The rigid frame is constrained in the z -direction of vibration by a spring of a small stiffness value, and laterally, by assumed gliding support (see Figure 9c). A passive damping force, an active

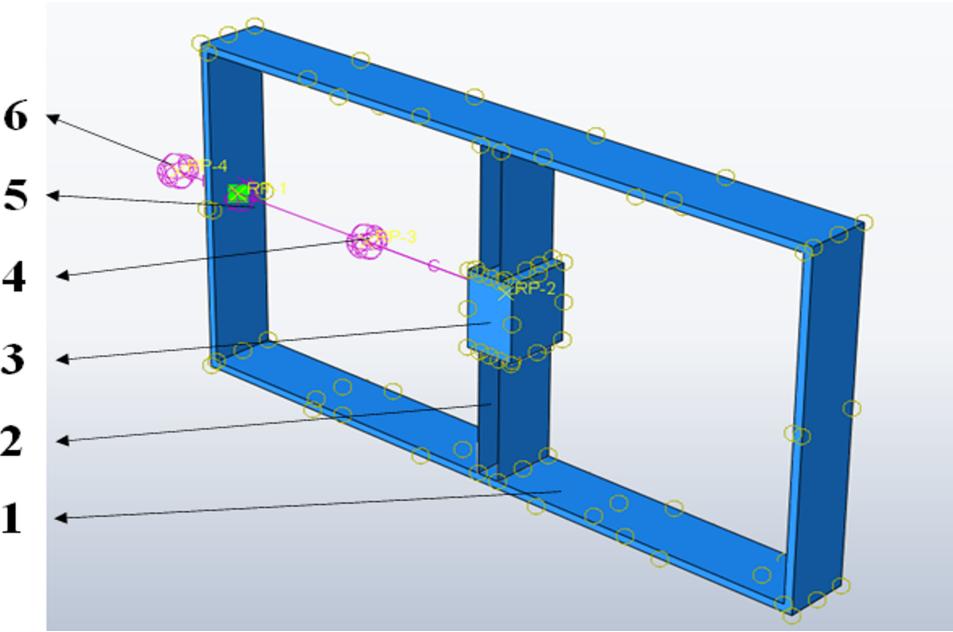


Figure 12: The dynamic vibration system: (1) the rigid frame, (2) the leaf springs, (3) the mass oscillator, (4) the damper, (5) the active force, and (6) the spring.

damping force (that result from the gain factor g) and elastic forces (from the leaf springs) apply on the mass oscillator. Dashpot elements are used for the damper and gain to model velocity-dependent forces. The damper represented by DASHPOT2 element introduces a damping force as a function of the relative velocity between the rigid frame and the mass oscillator, the active damping force due to gain is modelled as a function of the absolute velocity of the mass oscillator through DASHPOT1 element, and the spring is represented by SPRING1 element in ABAQUS. A Latin Hypercube sampling is completed with 98 runs for parameters values partially shown in Table 4 for which the damping ratio of the system is calculated.

Table 4: A partial parameterized input and corresponding numerical results

Case	Control Parameter x	Calibration Parameter θ_c	Design Parameter θ_d	System Response y
Variable unit	Mass (kg)	Elastic Modulus (N/m ²)	Gain factor (Ns/m)	Overall damping ratio (-)
1	0.9625	54037300000	11.5	0.0979
2	0.8175	58698200000	46.5	0.217
3	0.9525	72098300000	76.5	0.268
4	0.7275	70350500000	80.5	0.3496
5	1.0125	71515700000	73.5	0.2483
6	1.0875	64233000000	10.5	0.0815
...
93	1.0575	72389600000	54.5	0.1855
94	0.8775	68311300000	91.5	0.3621
95	0.7675	56950400000	19.5	0.1326
96	0.7525	71807000000	83.5	0.3489
97	0.8125	68893900000	27.5	0.1392
98	1.0525	48793800000	34.5	0.1679

4.4 Application of DCTO to vibration isolation design

Since our goal is to minimize the damping ratio, we set our target outcomes y_t to be 0 across a range of oscillator masses. Specifically, we set a grid of size 8 over the range of oscillator masses present in the simulation and experimental data, with target outcome 0 for each point in that grid. We define our prior GP surrogate for the FE model using a mean function found via degree-2 polynomial regression on the available FE runs. For the hyperparameters of the surrogate's covariance function, we estimate them as MLEs using the quasi-Newton BFGS method (Fletcher, 2013). We perform 10,000 iterations of MCMC using this surrogate and set of target observations, of which the first half are discarded as burn-in. The convergence of the resulting MCMC chains is assessed both visually and using the Gelman-Rubin statistic (≈ 1.01 and 1.001 for calibration and design respectively), Gelman and Rubin, 1992).

The total wall time required for the MCMC to complete DCTO in this case was 94 seconds (on a laptop with an Intel Core i7-9750H CPU and 16GB of RAM). The posterior distributions of the calibration and design inputs are shown in Figure 13. Strong Bayesian learning has occurred, particularly for the design input. The posterior distribution of the elastic modulus for the system assigns high likelihood to the expected value of $6.2\text{e}10$, with a posterior mean of $6.188\text{e}10$. For comparison with our design results, we also apply the NSGA-II algorithm (Deb et al., 2002), a gradient-free genetic algorithm, to the trained GP model surrogate. We use 100 generations and a population size 50, taking a total of 48 seconds of computation

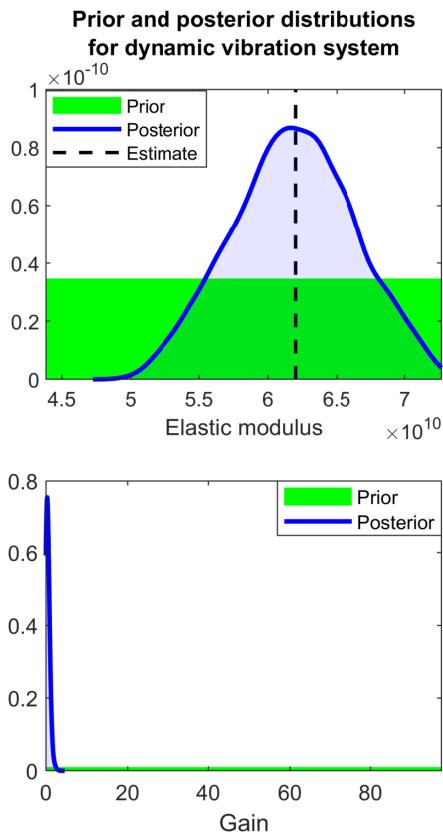


Figure 13: The posterior distributions of the calibration and design inputs, respectively, along with their (uniform) priors. The very narrow posterior distribution of gain is concentrated at the minimum of its support.

(wall time). Whereas our method performs both calibration and design, NSGA-II cannot be used for calibration, and so we apply it to a model calibrated with a point estimate (the posterior mean) of elastic modulus from our method’s results. The results of NSGA-II agree with our own, in finding the optimal gain setting to be 0.

We also use the surrogate model to estimate also the posterior predictive distribution of the system after DCTO. Figure 14 shows the resulting posterior distributions of model output at various levels of oscillator mass, along with the distributions of both experimental and simulator system output. For comparison, the figure also includes the output of the surrogate model using the posterior mean of elastic modulus along with the NSGA-II estimate of optimal gain. Note that the predicted model outputs fall at the bottom of the ranges of observed model outputs across the domain of oscillator masses, implying a successful design outcome for the system has been achieved.

5 Conclusion

DCTO provides a method for generalizing the KOH framework for model calibration to include design. The result secures the benefits of KOH both for calibration and for design. This includes the ability to quantify uncertainty remaining in the true value of the calibration parameter, the optimal settings for the design input, and the resulting model output. DCTO provides a computationally efficient method of propagating the uncertainties remaining from KOH calibration through the design procedure. In

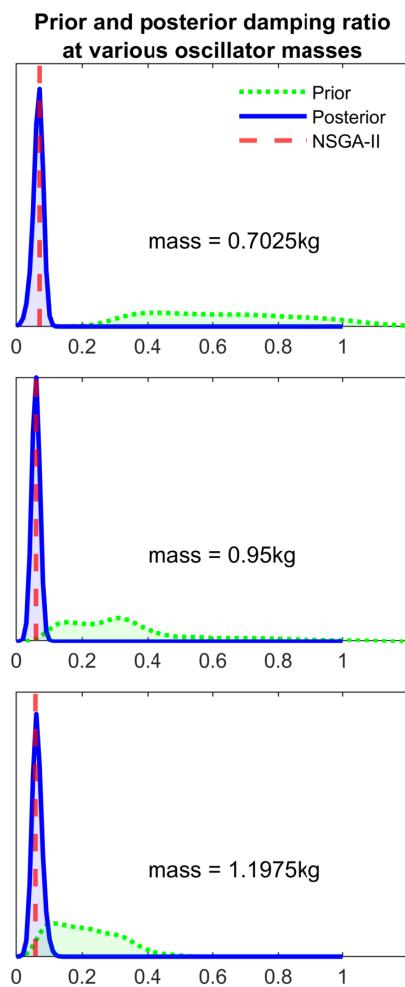


Figure 14: The posterior distributions of the model output at three different levels of the operational domain, along with their prior distributions. The posteriors constitute a notable performance increase over the priors.

the case when observations of the real system can be carried out sequentially at adaptively chosen locations, DCTO is robust to model misspecification where the calibration parameter is functionally dependent on the value of the design input and the model fails to reflect this. In such a case, if the functional form of the dependence of θ_c on θ_d is of interest, then state-aware calibration should be used. However, if one only wishes to estimate the calibration parameter at the optimal design settings, then DCTO provides a means of doing so. In this application, DCTO with adaptive sampling uses information from both the sequentially-performed observations of the real system and from the existing computer model to identify new sampling locations. Future work on this subject will include pairing adaptive sampling DCTO with other methodologies for selecting new sampling locations, such as EGO and SUR.

References

- Atamturktur, S. and D. A. Brown (2015). State-aware calibration for inferring systematic bias in computer models of complex systems. *NAFEMS World Congress Proceedings, June 21-24*.
- Atamturktur, S., J. Hegenderfer, B. Williams, M. Egeberg, R. A. Lebensohn, and C. Unal (2017). Mechanics of advanced materials and structures: a resource allocation framework for experiment-based validation of numerical models. *Mechanics of Advanced Materials and Structures* 22(8), 641–654.

Bayarri, M. J., J. O. Berger, and J. Cafeo (2007a). Computer model validation with functional output. *The Annals of Statistics* 35, 1874–1906.

Bayarri, M. J., J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu (2007b). A framework for validation of computer models. *Technometrics* 49(2), 138–154.

Binois, M., V. Picheny, P. Taillardier, and A. Habbal (2019, feb). The Kalai-Smorodinski solution for many-objective Bayesian optimization.

Bonyadi, M. R. and Z. Michalewicz (2017, mar). Particle swarm optimization for single objective continuous space problems: A review.

Branke, J., K. Deb, K. Miettinen, and R. Slowinski (Eds.) (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Volume 5252 LNCS. Springer.

Brochu, E., V. M. Cora, and N. de Freitas (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning.

Brown, D. A. and S. Atamturktur (2018). Nonparametric functional calibration of computer models. *Statistica Sinica* 28(2), 721–742.

Brynjarsdóttir, J. and A. O'Hagan (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems* 30(11).

- Chevalier, C., J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet (2014). Fast Parallel Kriging-Based Stepwise Uncertainty Reduction With Application to the Identification of an Excursion Set. *Technometrics* 56(4).
- Dean, A., D. Voss, and D. Draguljić (2017). *Response Surface Methodology*, pp. 565–614. Cham: Springer International Publishing.
- Deb, K. and H. Gupta (2006, dec). Introducing robustness in multi-objective optimization. *Evolutionary Computation* 14(4), 463–494.
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197.
- Ezzat, A. A., A. Pourhabib, and Y. Ding (2018, jul). Sequential Design for Functional Calibration of Computer Models. *Technometrics* 60(3), 286–296.
- Fletcher, R. (2013). *Practical methods of optimization*. John Wiley & Sons.
- Gelfand, A. E. and A. F. M. Smith (1990, jun). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85(410), 398–409.
- Gelman, A. and D. B. Rubin (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4), 457–472.

Geman, D. and B. Jedynak (1996). An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(1), 1–14.

Higdon, D., M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing* 26(2), 448–466.

Jones, D. R., M. Schonlau, and W. J. Welch (1998). Efficient Global Optimization of Expensive Black-Box Functions. Technical report.

Kennedy, M. C. and A. O'Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B* 63(3), 425–464.

Kim, M., T. Hiroyasu, M. Miki, and S. Watanabe (2004). SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3242, 742–751.

Lee, K. Y. and M. A. El-Sharkawi (2007, jun). *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*. John Wiley and Sons.

Liu, F., M. J. Bayarri, and J. O. Berger (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis* 4(1), 119–150.

Mason, K., J. Duggan, and E. Howley (2017, dec). Multi-objective dynamic economic

emission dispatch using particle swarm optimisation variants. *Neurocomputing* 270, 188–197.

Miguel Hernández-Lobato, J., M. A. Gelbart, R. P. Adams, M. W. Hoffman, Z. Ghahramani, J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani Hernández-Lobato (2016). A General Framework for Constrained Bayesian Optimization using Information-based Search. Technical report.

Mockus, J., V. Tiesis, and A. Zilinskas (1978). The application of bayesian methods for seeking the extremum. *Towards global optimization* 2(117-129), 2.

Nocedal, J. and S. J. Wright (2006). *Numerical optimization*. Springer.

Olalotiti-Lawal, F. and A. Datta-Gupta (2018). A multiobjective markov chain monte carlo approach for history matching and uncertainty quantification. *Journal of Petroleum Science and Engineering* 166, 759–777.

Paulo, R., G. García-Donato, and J. Palomo (2012). Calibration of computer models with multivariate output. *Computational Statistics and Data Analysis* 56, 3959–3974.

Picheny, V. (2015). Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing* 25(6), 1265–1280.

Picheny, V., M. Binois, and A. Habbal (2019, jan). A Bayesian optimization approach to find Nash equilibria. *Journal of Global Optimization* 73(1), 171–192.

Regis, R. G. and C. A. Shoemaker (2004). Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Transactions on Evolutionary Computation* 8(5), 490–505.

Robert, C. P. and G. Casella (2004). Monte Carlo Optimization. In *Monte Carlo Statistical Methods* (2 ed.), Chapter 5, pp. 157–204. New York, NY: Springer New York.

Villemonteix, J., E. Vazquez, and E. Walter (2009, aug). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization* 44(4), 509–534.

Williams, B., D. Higdon, J. Gattiker, L. Moore, M. McKay, and S. Keller-McNulty (2006). Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis* 1(4), 765–792.

Zhou, A., B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang (2011, mar). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1(1), 32–49.

Appendix: Validation of DCTO

In order to validate the performance of simultaneous calibration and design under DCTO against a two-step calibration and design approach, we use the system described by Equations (4,5,6) where the value of calibration parameter does not depend on the value of the design input. We employ a space-filling design of observations of the “true” system under both approaches. For the two-step approach, we carry apply traditional KOH calibration of θ_c , followed by a second step using the KOH framework for design, obtaining a distribution of θ_d . The first step is thus essentially DCTO with $\mathbf{x}_t, \mathbf{y}_t$ as empty (null) vectors, and the second step uses the distributions obtained in the first step to estimate θ_d . Thus, the comparison between the unified approach, i.e. DCTO, and the two-step calibration and design approach shows the difference between DCTO and performing design on a system which has been calibrated using traditional methods and serves as validation of the former.

Figure 15 shows the results of DCTO and two-step calibration and design for f_0 , the case of no discrepancy. The two methods deliver comparable results, illustrating that extending calibration to include design does not undermine the performance of either task. Strong Bayesian learning has occurred for both parameters, in that the posterior distributions of θ_c, θ_d are peaked around their true and optimal values, respectively. KOH gives a similar posterior for θ_c , showing that the expansion of DCTO to undertake design has not interfered with its calibration performance. The skew apparent in the posterior distributions of θ_d occur in all of the results gathered

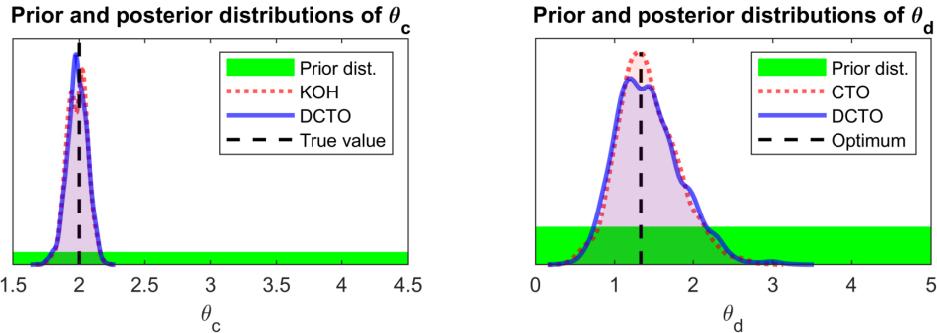


Figure 15: Prior and posterior distributions of the calibration parameter θ_c and design parameter θ_d , along with their true/optimal values, for DCTO and two-step calibration and design carried out when there is no discrepancy between the true system described by Equation (2) (the case of no discrepancy) and the computer model.

here, and is likely due to the shape of the objective function f , which is much more informative below θ_d than above it in that it increases sharply for $t_d < \theta_d$ and increases much more gently for $t_d > \theta_d$.

We performed each procedure 30 times on each of the seven different discrepancy situations (no discrepancy, and a large and small version of each of three discrepancies). The results are summarized in Table 5. The upper table gives the sample mean, over the thirty runs, of the marginal posterior variance of each of θ_c and θ_d . The two procedures generate extremely similar outcomes with respect to θ_c . However, the posterior variance for θ_d under DCTO is slightly higher than that under two-step calibration and design in each of the seven cases considered. This is due to the fact that DCTO includes remaining uncertainty about the values of the hyperparameters of the discrepancy GP $\delta(\cdot)$. By contrast, design after calibration uses point estimates of those hyperparameters and thus achieves narrower posterior distributions due to

	Posterior θ_c var.		Posterior θ_d var.	
Discrepancy	DCTO	KOH+design	DCTO	KOH+design
f_0 (no discrepancy)	0.00633	0.00619	0.145	0.129
$f_1, a = 1.5$	0.0140	0.0137	0.149	0.129
$f_1, a = 3.5$	0.0141	0.0140	0.149	0.131
$f_2, a = .15, b = .075$	0.0143	0.0141	0.135	0.116
$f_2, a = .65, b = .075$	0.0609	0.0608	0.0804	0.0731
$f_3, a = .055, b = 0$	0.0134	0.0135	0.0882	0.0743
$f_3, a = .055, b = .1$	0.0143	0.0142	0.0945	0.0814

	$\widehat{\theta}_c$ RMSE		$\widehat{\theta}_d$ RMSE	
Discrepancy	DCTO	KOH+CTO	DCTO	KOH+CTO
f_0 (no discrepancy)	0.0790	0.0795	0.120	0.121
$f_1, a = 1.5$	0.0955	0.0956	0.149	0.144
$f_1, a = 3.5$	0.137	0.139	0.209	0.209
$f_2, a = .15, b = .075$	0.109	0.106	0.130	0.127
$f_2, a = .65, b = .075$	0.158	0.155	0.123	0.121
$f_3, a = .055, b = 0$	0.294	0.292	0.0919	0.0919
$f_3, a = .055, b = .1$	0.279	0.281	0.0995	0.0990

Table 5: Posterior variance and root mean square error (RMSE) for the calibration variable θ_c and the design variable θ_d under both DCTO and two-step calibration and design (KOH+design). The estimator $\widehat{\theta}_i$ is the posterior mean of t_i for $i = c, d$. For each f_i , a and b control the size of the discrepancy as specified in Equations (4,5,6).

excluding this source of uncertainty. The lower table gives the root mean square errors (RMSEs) for the posterior means of θ_c and θ_d , using their true value of 2 for θ_c and optimal value $4/3$ for θ_d in discrepancy cases 0, 1, 2 and optimal value 1 for discrepancy 3. Again we see very similar outcomes in the two procedures for both parameters. In all cases but one, DCTO has slightly higher RMSE for θ_d than does design after KOH calibration. This is to be expected given the above-mentioned wider posterior distributions of θ_d under DCTO. In general, we see that the DCTO expansion of the KOH framework achieves the expected similar results to KOH calibration followed by a design step, though with wider posterior distributions of the design input, because the DCTO approach properly includes in the design step a source of uncertainty (hyperparameters of hte discrepancy GP $\delta(\cdot)$) ignored by the traditional two-step approach.

Conclusion

Carl Ehrett
Watt Family Innovation Center,
Clemson University

1 Benefits

The research presented in the previous chapters addresses two distinct desiderata related to model-assisted design. Firstly, there is the desideratum of undertaking model-assisted design in a way that accounts for all forms of uncertainty – uncertainty due to the model inputs, uncertainty due to the stochastic nature of the objective function, and/or uncertainty due to observation error of the outputs. All of these sources of uncertainty can be modeled and included in the Bayesian framework used to employ our methodology. The resulting posterior distribution of the design inputs quantifies uncertainty as to what inputs could lead to optimal system behavior. The corresponding posterior predictive distributions quantifies uncertainty as to that resulting system behavior, including uncertainty of the entire Pareto front of the system. In contrast with approaches such as that of Olalotiti-Lawal and Datta-Gupta (2015), who provide similar uncertainty quantification of design input settings using a distribution they contrive, our method provides the uncertainty in a posterior distribution that is directly dictated by what is known about the computer model itself,

by one's prior knowledge about the appropriate design settings, and by the priorities of decision-makers.

Our method furthermore evades the need to be able to evaluate the objective function adaptively. This requirement is a limitation shared by most other Bayesian optimization (BO) methods. In this way, our method may be employed in scenarios where researchers are confined to the usage of pre-existing data sets, or in scenarios where the experimental design used for data-gathering must satisfy priorities other than that of engineering design.

Secondly, there is the desideratum of unifying procedures for calibration and design. Typically these two tasks are undertaken separately; a model would be calibrated and then the calibrated model would be put to use for model-assisted design. However, design priorities arise for models that stand in need of calibration, and wedging the frameworks for calibration and design allows for a single use of a dataset to satisfy both sets of goals.

2 Summary of chapter two

In chapter 2, we focused on model-assisted design, assuming there the possession of an already-calibrated computer model. We there considered the problem of engineering design from the perspective of, and using tools from the field of, computer model calibration under uncertainty. Specifically, we approached model-assisted design from the Bayesian framework for model calibration developed by Kennedy and

O'Hagan (2001), which we here refer to as KOH. We used this approach to undertake model-assisted design on a multi-objective system, quantifying remaining uncertainty regarding the optimal values of the design inputs and the resulting model output.

To do this, we relied on a method we call Counterfactual Bayes. Ordinarily, under KOH, one has access to a set of experimental observations that one uses to calibrate a model, so that the model output approximates those observations well. In the case of design, we wish to induce the model output to be *optimal*; we do not have access to any particular set of real experimental observations that are relevant to that goal. However, we argued, we may apply the KOH framework to design by reasoning using hypothetical – i.e. counterfactual – observations which would only occur if the design settings *were* optimal. By selecting such *target outcomes*, we are able to apply KOH and thereby discover distributions of design inputs that induce the model to approximate the hypothetical target outcomes. Since the target outcomes could be observed only when the design settings are optimal, our posterior distributions of design inputs achieving the target outcomes are *de facto* distributions of *optimal* design inputs. Thus we showed that KOH can be enlisted through Counterfactual Bayes as a powerful methodology for model-assisted design with quantification of all relevant uncertainty. In our discussion we included guidance as to the appropriate choice of target outcomes. Specifically, we demonstrated how, with little added computational cost, an initial “rough estimate” of the system Pareto front can be generated and used to select effective target outcomes.

To accommodate models of high computational expense, we employed GP surrogate models. We offered guidance on the selection of prior distributions for model inputs and for GP hyperparameters, as there are some aspects of these choices that are particular to the use of KOH with target outcomes. We then described an algorithm for the repeated application of our methodology in order to estimate not merely the optimal design settings with respect to some one goal, but instead to estimate the entire Pareto front for the system of interest. This allows decision-makers full flexibility in selecting a design to meet any set of priorities.

We demonstrated our methodology on a simulated example, where we were able to display the results of the procedure using the known optimum of the system. We also demonstrated our methodology, including the algorithm for full Pareto front estimation, on an application of material design for a wind turbine blade. We included a comparison showing that our resulting estimated Pareto front agrees with an estimate provided by NSGA-II (Deb et al., 2002), and that our estimate (unlike that of NSGA-II) is able to provide credible bands quantifying uncertainty remaining in the Pareto front location.

We concluded that our method captures much of what is attractive in other BO methods, without requiring the ability to sample the objective function adaptively. Our method thus is a useful addition to the set of tools available to a researcher wishing to undertake model-assisted multi-objective design under uncertainty.

3 Summary of chapter three

In chapter three, we broadened our methodology to include simultaneous calibration and design. Whereas these two procedures are typically carried out separately, we show that our adaptation of the Bayesian KOH framework is able to accommodate both tasks in a unified framework. This unified framework is a computationally efficient method of undertaking calibration and design with quantification of relevant uncertainties.

In KOH, one calibrates a model using a set of experimental observations. If the model is computationally expensive, then one also uses a set of “observations” of the model instead of evaluating the model as part of the calibration procedure. Thus KOH weds experimental observations and model observations for the purpose of calibration. In chapter two, we replaced the model observations with target outcomes for the purpose of model-assisted design. In chapter three, we expand the KOH framework so that it weds all three of these: experimental observations, model observations, and target outcomes. We described the joint distribution of the concatenation of these three vectors, allowing for the exploration of the joint posterior distribution of the calibration and design parameters. We made clear in what way the resulting framework successfully separates calibration from design – so that, e.g., the posterior distribution of the calibration parameters would be drawn to their *true* values (rather than values that would optimally yield the target outcomes) and the posterior distribution of the design inputs would similarly be drawn to their optimal

values. We furthermore described the modularization (Liu et al., 2009) that we employ in order to prevent the calibration parameters from being unduly influenced by the target outcomes.

Having established our methodology for dual calibration and design, we demonstrated using a simulated data set, in which the “true” calibration parameter and optimal design input were known. In order to demonstrate the flexibility of the method and its robustness to model bias, we considered seven different versions of the simulated system: a version without model bias, and a small and large version of each of three different discrepancies of different forms. Using MCMC to explore the posterior distributions of the calibration and design inputs, we showed the resulting success of the methodology in each of these cases, while also noting the (expected) degradation in performance as model bias grows large.

To further demonstrate the usefulness of the method, we showed how it may be of special value when the calibration parameter is functionally dependent upon the design input and adaptive sampling of the high-fidelity model is possible. Toward this end, we considered another simulated system, in which such dependence occurs. We described an algorithm for wedding our dual calibration and design to an adaptive sampling strategy that induces the high-cost observations to be clustered in the region of greatest interest from the perspective of design. The result is that the model calibration is grounded on observations that occur when the value of the calibration parameter is near to its value at the optimal state of the system. We again

demonstrated the results of this approach using seven different discrepancy cases.

Finally, in addition to the simulated systems described above, we applied our methodology to the vibration isolation design application. We demonstrated the ability of our framework to identify plausible posterior distributions on the calibration parameter of leaf spring elastic modulus and the design parameter of the gain factor. Having demonstrated the results of our methodology, we concluded that our expanded alteration of the KOH framework is able to provide unified calibration and design with similarly unified uncertainty quantification. In this way the researcher sees not merely the marginal uncertainty of the calibration parameters and that of the design inputs, but also sees their joint posterior distribution and the associated uncertainty. The uncertainty of the posterior predictive distribution of model behavior is thus captured as well.

4 Recommendations

We have demonstrated the novel benefits of the present work’s method. However, it is worthwhile to note the areas in which the method may have limited applicability. In general, the weaknesses of the present approach are the weaknesses shared with the KOH framework generally. Specifically, when relying on GP surrogates, the computational infelicities of GPs apply. GPs may struggle with high-dimensional data, and also with large data sets. In general, to use a data set of size n for a GP, one must invert a matrix of size n . Happily, while the present approach shares the challenges

of KOH in this area, it can similarly partake in the large body of work devoted to alleviating or mitigating these problems. Works described in Chapter 1 such as that of Higdon et al. (2008), Bhat et al. (2010), Paulo et al. (2012), Drignei and Mourelatos (2012), Pratola et al. (2013), and Higdon et al. (2013) are all applicable to the present work to improve computational efficiency and stability.

A benefit of the present approach is that, in contrast with many other Bayesian approaches to model-assisted design, it does not require the ability to sample adaptively from the objective function. Conversely, however, when adaptive sampling is possible, it has appreciable benefits in reducing the total number of objective function evaluations needed for design (Jones et al., 1998; Vazquez and Bect, 2009; Bect et al., 2012; Chevalier et al., 2014). Nonetheless, as described in Chapter 3, these BO approaches do not accommodate the case in which simultaneous calibration and design are undertaken. The present approach may still be advantageous here, especially when, as in Chapter 3, there is a known or suspected functional dependency of the calibration parameters upon the design inputs.

When one's data allows for a GP surrogate of reasonable computational efficiency (or when one's objective function is of low enough computational efficiency that a surrogate is not required), and when one either wishes to undertake simultaneous calibration and design (either for efficiency or because of a suspected functional relationship of calibration parameters on design inputs) or else one is constrained not to employ adaptive sampling of the objective function, then we recommend the ap-

plication of the method presented in this work.

5 Future directions

We remarked above that the framework advocated here is compatible with broad efforts to increase the scalability of MCMC and of GPs. A fruitful next step for research in this area would be to effect such integration and to explore its effectiveness in relevant applications. This can take a number of forms, as there exist a wide array of strategies for improving the computational feasibility and stability of MCMC and of GPs in high-dimensional and large-data contexts. It would also be fruitful to apply the present approach to cases in which the design space contains proper subsets that are infeasible, by imposing regularization to constrain the solution space (e.g. as in Calvetti et al. (2014)). Similarly, whereas in the present work we focus on unconstrained problems, it would be instructive to combine our approach with available methods for applying GPs that respect known constraints (Golchi et al., 2015; Wang and Berger, 2016; Maatouk and Bay, 2017; Ding et al., 2019). Another potential area for future research is the union of our dual calibration and design approach with existing BO methods for selecting new sampling locations, such as the EGO approach of Jones et al. (1998) or the SUR approach of Vazquez and Bect (2009), Bect et al. (2012) and Chevalier et al. (2014).

References

- Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793.
- Bhat, K., Haran, M., and Goes, M. (2010). Computer model calibration with multivariate spatial output. In Chen, M.-H., Dey, D., Müller, P., Sun, D., and Ye, K., editors, *Frontiers of Statistical Decision Making and Bayesian Analysis: in Honor of James O. Berger*, pages 168–184. Springer, New York.
- Calvetti, D., Kaipio, J. P., and Somersalo, E. (2014). Inverse problems in the Bayesian framework. *Inverse Problems*, 30(11).
- Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014). Fast Parallel Kriging-Based Stepwise Uncertainty Reduction With Application to the Identification of an Excursion Set. *Technometrics*, 56(4).
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Ding, L., Mak, S., and Wu, C. F. J. (2019). BdryGP: a new Gaussian process model for incorporating boundary information.
- Drignei, D. and Mourelatos, Z. P. (2012). Parameter screening in statistical dynamic

computer model calibration using global sensitivities. *Journal of Mechanical Design, Transactions of the ASME*, 134(8).

Golchi, S., Bingham, D. R., Chipman, H., and Campbell, D. A. (2015). Monotone emulation of computer experiments. *SIAM-ASA Journal on Uncertainty Quantification*, 3(1):370–392.

Higdon, D., Gattiker, J., Lawrence, E., Jackson, C., Tobis, M., Pratola, M., Habib, S., Heitmann, K., and Price, S. (2013). Computer model calibration using the ensemble kalman filter. In *Technometrics*, volume 55, pages 488–500. Taylor & Francis Group.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. Technical report.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464.

Liu, F., Bayarri, M. J., and Berger, J. O. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150.

- Maatouk, H. and Bay, X. (2017). Gaussian Process Emulators for Computer Experiments with Inequality Constraints. *Mathematical Geosciences*, 49(5):557–582.
- Olalotiti-Lawal, F. and Datta-Gupta, A. (2015). A Multi-Objective Markov Chain Monte Carlo Approach for History Matching and Uncertainty Quantification. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.
- Paulo, R., García-Donato, G., and Palomo, J. (2012). Calibration of computer models with multivariate output. *Computational Statistics and Data Analysis*, 56:3959–3974.
- Pratola, M. T., Sain, S. R., Bingham, D., Wiltberger, M., and Rigler, E. J. (2013). Fast sequential computer model calibration of large nonstationary spatial-temporal processes. *Technometrics*, 55(2):232–242.
- Vazquez, E. and Bect, J. (2009). A sequential bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes*, 42(10):546–550.
- Wang, X. and Berger, J. O. (2016). Estimating shape constrained functions using Gaussian processes. *SIAM-ASA Journal on Uncertainty Quantification*, 4(1):1–25.