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 an 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., $\text{argmin}_{\boldsymbol{\theta}}\, f_\alpha(\mathbf{x}, \boldsymbol{\theta}) = \text{argmin}_{\boldsymbol{\theta}} \|\mathbf{y}_t - f_\alpha(\mathbf{x}, \boldsymbol{\theta})\|$ 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 $\mu : \mathcal{D} \to \mathbb{R}$ and $C : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$, 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

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

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 $i = 1, \ldots, n$. 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:

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

where $\beta_k,\ k = 1,\ldots,p+q$, 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 $j = 1,\ldots,p+q$, 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 $\boldsymbol{\beta}^\eta$ 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 $\boldsymbol{\beta}^\eta$. 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 $\boldsymbol{\eta} = (\eta(\mathbf{x}_1, \mathbf{t}_1), \cdots, \eta(\mathbf{x}_n, \mathbf{t}_n))^T$, target outcomes $\mathbf{y}_t = (y_t(\mathbf{x}_{n+1}), \cdots, y_t(\mathbf{x}_{n+m}))^T$, and $D = (\boldsymbol{\eta}^T, \mathbf{y}_t^T)^T$. Following the counterfactual framework, we take the distribution of $D|\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \widehat{\lambda_\eta}, \widehat{\boldsymbol{\rho}^\eta}$ 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 $\boldsymbol{x}$, $\pi(\boldsymbol{x}) \propto 1$, 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

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

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 $\boldsymbol{\sigma}^2 = s(1, 1, \ldots, 1)^T$ 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 $\sigma_i^2 \sim$ gamma(4/1/8) (for example) for $i = 1, \ldots, m$, 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-

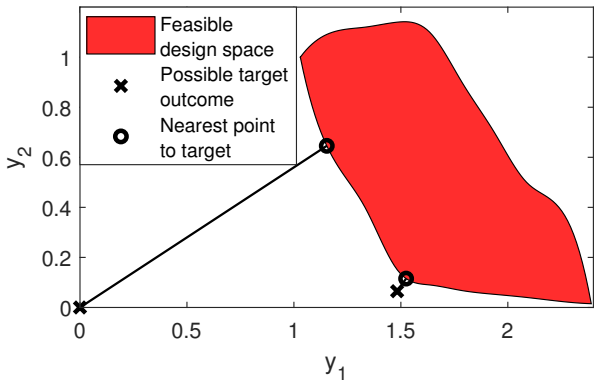

Figure 1: Two choices of target outcomes for CTO, drawing the posterior predictive distribution to two different regions of the feasible design space.

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_5)$ 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

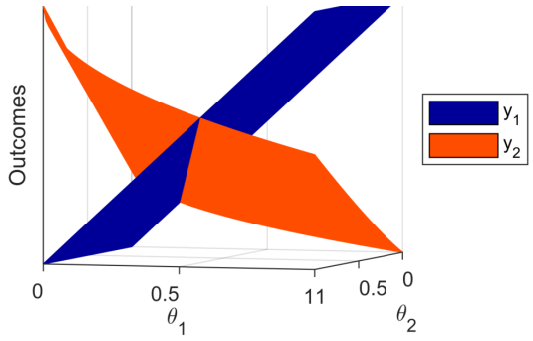

Figure 2: True two-dimensional profile outputs of the five-dimensional simulated example model.

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 $\boldsymbol{\theta}$, 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

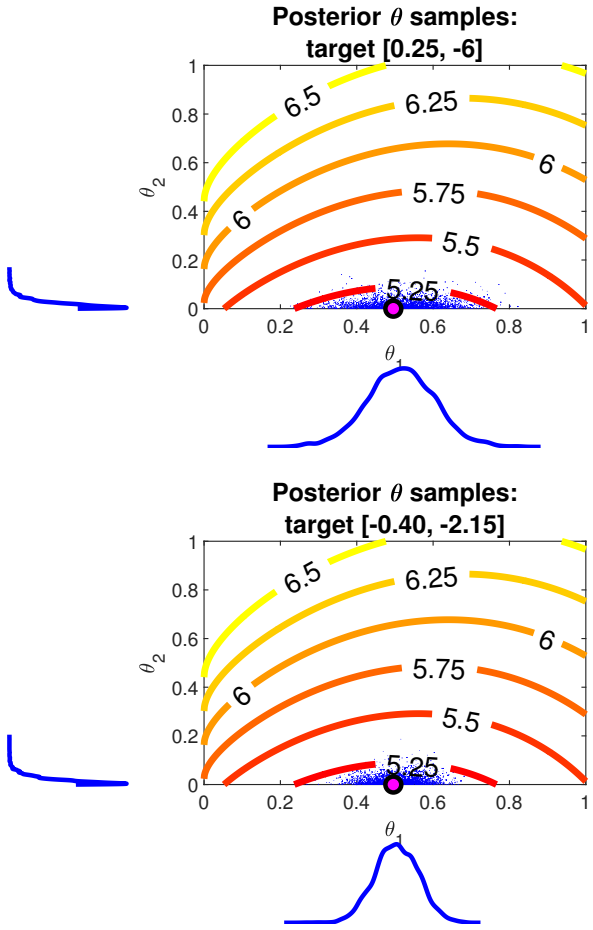

Figure 3: Posterior draws from CTO in the simulated example using an arbitrary (non-feasible) point as a target (top) and using an updated target designed to lie two standard deviations from the Pareto front, in the direction of the original target (bottom). The contours show, for each point in the design 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 $\lambda_\eta, \boldsymbol{\beta}^\eta$ are estimated via maximum likelihood using only the finite element model output. We used `fmincon()` in `MATLAB` (MATLAB, 2017) to maximize (with $D = \boldsymbol{\eta}$) over the joint (four-dimensional) support of $\boldsymbol{\beta}^\eta, \lambda_\eta$. 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 $\boldsymbol{\sigma}^2 = 5 \times 10^7 \cdot (1, 1, 1)$ 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 = 0.75m, twist = 0.09 rad, cost = \$129.68) as the target outcome, replicated to be constant as a

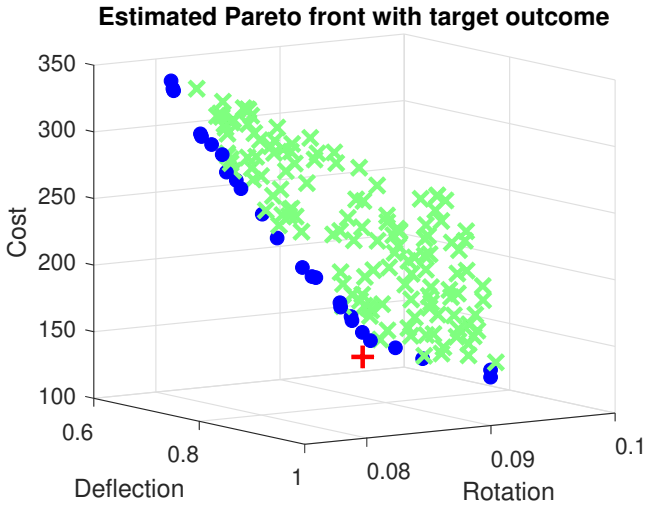

Figure 4: Each x is a dominated design drawn from the predictive distribution through preliminary CTO. The dots indicate the estimated Pareto front. The plus sign is the target selected as the performance objective in our proposed design approach.

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 $\boldsymbol{\sigma}^2$ an 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

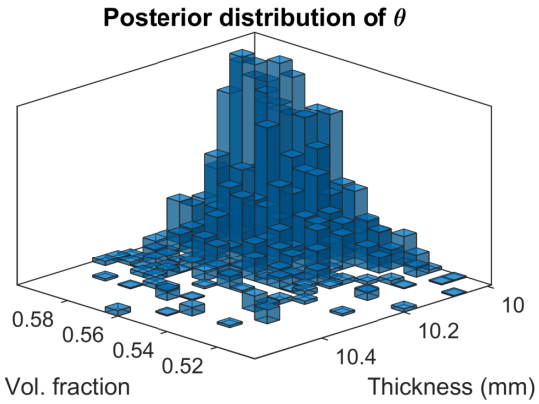

Figure 5: Histogram showing the posterior distribution from CTO in the wind turbine blade system. The prior is uniform over $[0.1, 0.6] \times [10, 25]$.

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 $(0.753\text{m}, 0.091 \text{ rads}, \$206.58/\text{m}^2)$,

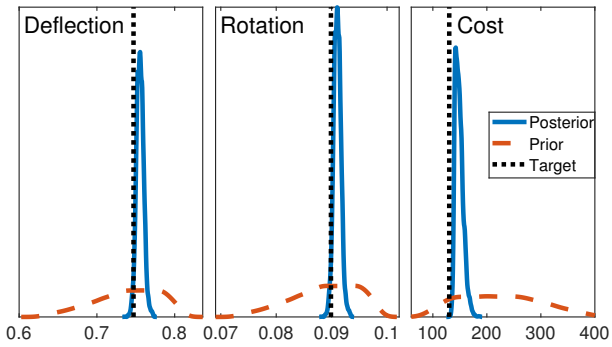

Figure 6: Approximate prior and posterior marginal predictive densities for each of the three outputs in the turbine blade design problem.

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-

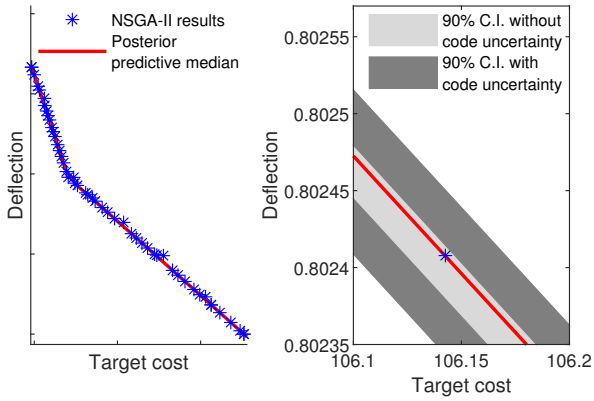

Figure 7: The estimated Pareto front of the wind turbine blade system with quantified uncertainties, along with NSGA-II estimation of the front. The light gray shows the 90% credible interval for the front without code uncertainty (i.e., treating the emulator as perfect); the dark gray extends the credible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

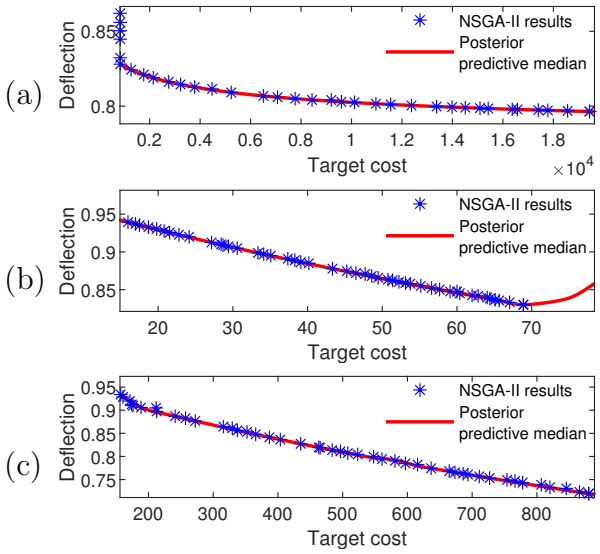

Figure 8: Estimated Pareto front for multiple wind turbine blade systems with respect to a variety of design spaces, along with NSGA-II estimation of the fronts. 95% credible intervals are too small to be visible.

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 $\boldsymbol{\theta}$ 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 $\boldsymbol{\theta}$ 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 multiobjective 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.

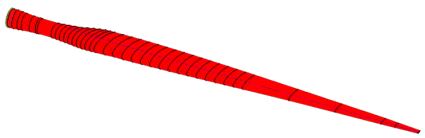

Figure 9: CX-100 blade model created in NuMAD for ANSYS input file generation and finite-element analysis.

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.

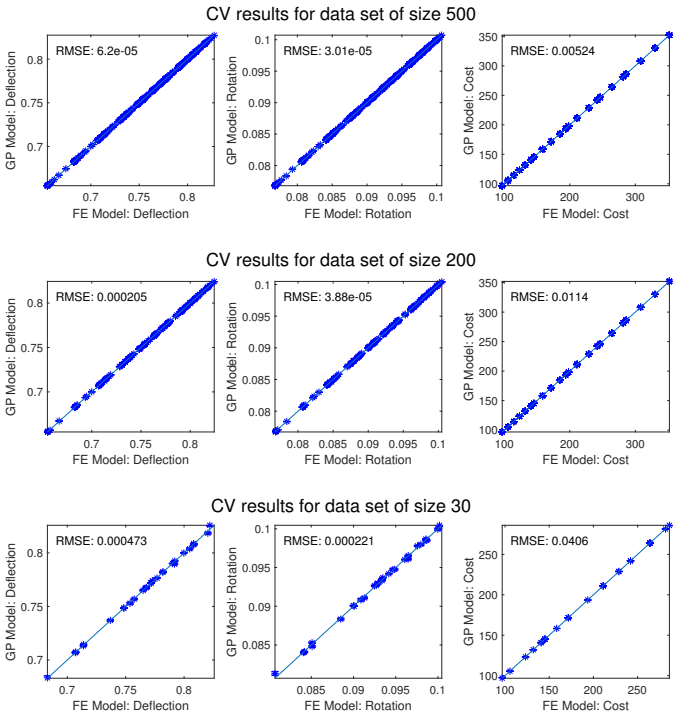

Figure 10: Results of 10-fold cross validation of the GP emulator used for the wind turbine application.