Coupling material and mechanical design processes via computer model calibration

Carl Ehrett*
School of Mathematical and Statistical Sciences, Clemson University,
D. Andrew Brown
School of Mathematical and Statistical Sciences, Clemson University,
Evan Chodora
Department of Mechanical Engineering, Clemson University,
Mingzhe Jiang
Department of Chemical and Biomolecular Engineering, Clemson University,
Christopher Kitchens
Department of Chemical and Biomolecular Engineering, Clemson University,
and
Sez Atamturktur
Department of Architectural Engineering, Pennsylvania State University

July 9, 2019

Abstract

Computer model calibration typically operates by choosing parameter values in a computer model so that the model output faithfully predicts reality. By using desired performance targets in place of observed data, we show that calibration techniques can be repurposed to wed engineering and material design, two processes that are traditionally carried out separately. This allows materials to be designed with specific engineering targets in mind while quantifying the associated sources of uncertainty. We demonstrate our proposed approach by "calibrating" material design settings to performance targets for a wind turbine blade and by estimating the system's Pareto front with quantified uncertainties.

Keywords: Gaussian processes, material design, optimization, Pareto optimality, Uncertainty quantification, wind turbines

*The authors gratefully acknowledge *please remember to list all relevant funding sources in the unblinded version*

1 Introduction

Real-world optimization problems typically involve multiple objectives. This is particularly true in the design of engineering systems, where multiple performance outcomes are balanced against budgetary constraints. Among the complexities involved in optimizing over multiple objectives is the effect of uncertainties in various aspects of the problem. Design is guided by models known to be imperfect, systems are built using materials with uncertainty regarding their material 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 given design problem.

In traditional engineering design, one designs a system after choosing a material with appropriate properties for the project from a database of known materials. As a result, the design of the system is constrained by the initial material selection. By coupling material discovery and engineering system design, we can combine these two traditionally separate processes under the umbrella of a unified multiple objective optimization problem.

In this paper, we cast the engineering design problem in the framework of computer model calibration. In traditional calibration, one aligns the computer model output to observations of the 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.

In addition to optimizing to a specific set of targets, in many situations it is helpful to have a comprehensive picture of optimal outcomes without committing to specific targets. More generally, in a system with multiple outputs, one wants to estimate the Pareto front

of the design space; i.e., the set of points in the design space that are *Pareto optimal*. A point is Pareto optimal if and only if, in order to improve any one of its elements, some other element must be made worse off. In a system in which there is a trade-off between cost and performance, for example, which region of the Pareto front is considered most desirable will depend upon the budgetary constraints of the project. We show that our proposed methodology can be used to produce "Pareto bands" which estimate the Pareto front of a system with quantified uncertainties.

Our proposed methodology uses the Bayesian framework first established as a means for computer model calibration by Kennedy and O'Hagan (2001). This area is furthered by Higdon et al. (2004), who undertake model calibration with quantification of the related uncertainty. They explicitly incorporate uncertainty regarding the computer model input, the bias of the computer model, and uncertainty due to observation error. The approach of Higdon et al. (2004) is further refined and exemplified by Williams et al. (2006). 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 (using the same training data). Bayarri et al. (2007) apply this methodology to functional data 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 when undertaking

calibration.

Common to those approaches is the conception of calibration as using real observations to get a posterior distribution on unknown parameters $\boldsymbol{\theta}$ such that the posterior predictive distribution of the model approximates the real system. By contrast, in our proposed methodology, we use artificial observations (representing our design targets) to obtain a posterior distribution on design variables $\boldsymbol{\theta}$ such that the posterior predictive distribution of the model approaches our design targets. Repeated applications of the procedure can be used to produce the more thorough "Pareto bands" which estimate the Pareto front with quantified uncertainties.

We apply our proposed methodology both to a proof-of-concept example and to the problem of finding material design settings to optimize the performance and cost of a wind turbine blade of fixed outer geometry. The wind turbine blade in question is to be constructed using a composite material. One design variable targeted for optimization is the *volume fraction*, which is the ratio of the *filler* to *matrix* used in the composite material. In a composite, the matrix holds the filler together; an example would be concrete, in which a filler of loose stones is combined with a matrix of cement. Another design variable is the thickness (in mm) of the shear web used in the blade. Our material design goal is to reduce the cost per square meter of the composite material, the angle of twist (in radians) of the blade when under load, and the deflection (in meters) of the blade tip when under load.

In Section 2, we review the calibration framework that serves as the basis for our proposed design optimization approach. In Sections 3 and 4, we apply our proposed methodology to the example involving simulated data and to wind turbine blade design. In Section

4, we show how our approach can be used to produce an estimate of the Pareto front of the wind turbine blade system while quantifying associated uncertainty. Section 5 concludes with discussion of the results and thoughts about future directions.

2 Calibration for design

2.1 Gaussian process emulators for calibration

In this work, when an emulator is needed we assume the use of a Gaussian process (GP) emulator. Just as a multivariate Gaussian random variable is characterized by its mean vector and covariance matrix, a GP is fully characterized by its mean function $\mu : D \to \mathbb{R}$ and covariance function $C : D \times D \to \mathbb{R}$, where D is the domain of the process. Thus for any points $\mathbf{x}, \mathbf{y}$ in the domain of the GP, $\mu(\mathbf{x})$ gives the mean of the GP at $\mathbf{x}$, and $C(\mathbf{x}, \mathbf{y})$ gives the covariance between the values of the GP at points $\mathbf{x}$ and $\mathbf{y}$. The distribution of the process at any finite number of points is multivariate Gaussian with mean vector and covariance matrix determined by $\mu(\cdot)$ and $C(\cdot, \cdot)$. In principle, model calibration need not rely on a GP emulator, or any other sort of emulator; one could complete a full Bayesian analysis via Markov chain Monte Carlo (MCMC; Gelfand and Smith, 1990) by running the relevant computer model at each iteration of the chain (see e.g. Hemez and Atamturktur, 2011). In Section 3 we assume fast-running computer code for the simulated example. However, computer models are frequently too computationally expensive to allow for such expenditure (Van Buren et al., 2013, 2014). Instead, a computationally tractable emulator can be constructed using a sample of output from the computer model.

GPs are popular prior distributions on computer model output for three reasons. Firstly, because their use does not require detailed foreknowledge of the model function's parametric form. Secondly, GPs easily interpolate the computer model output, which is attractive when the computer model is deterministic and hence free of measurement error. This is usually the case, although some attention in model calibration (e.g., Pratola and Chkrebtii, 2018) has focused specifically on stochastic computer models. Thirdly, GPs facilitate uncertainty quantification through the variance of the posterior GP. This section provides brief background on Gaussian processes and their use in regression broadly, and in computer model calibration specifically.

The use of GPs to produce a computationally efficient predictor of expensive computer code given observations of code output at $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ is advocated by Sacks et al. (1989) and explored at length by Santner et al. (2003). Since computer code is typically deterministic, these applications differ from the focus of O'Hagan (1978) in that the updated GP is induced to interpolate the computer output $\boldsymbol{\eta} = (\eta(\mathbf{x}_1), \cdots, \eta(\mathbf{x}_n))^T$. Kennedy and O'Hagan (2001) use 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 both numerical and graphical diagnostic techniques for assessing when a GP emulator of a computer model is successful, as well as likely causes of poor diagnostic results. While most work in the area of GP emulation uses stationary covariance functions and quantitative inputs, efforts have been made to branch away from these core assumptions. Gramacy and Lee (2008) use treed partitioning to deal with a nonstationary computer model. Qian et al. (2008) explore methods for using GP emulators that include

both quantitative and qualitative inputs.

Whether or not an emulator is used, in the framework implemented here one may consider a computer model to be of the form $\eta(\mathbf{x}, \boldsymbol{\theta})$, where $(\mathbf{x}, \boldsymbol{\theta})$ comprise all inputs to the model. The vector $\boldsymbol{\theta}$ denotes the collection of design variables. The vector $\mathbf{x}$ is the collection of all other inputs that are known and/or under the control of the researcher. We call $\mathbf{x}$ the *control inputs*. Thus, the model is

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

where $y(\mathbf{x})$ describes a model outcome at control inputs $\mathbf{x}$, $\delta(\cdot)$ describes the model discrepancy (the systematic bias of the model as an estimate of the response) and $\epsilon(\cdot)$ is a mean-zero error, often assumed to be i.i.d. Gaussian.

To employ an emulator, suppose that we have inputs $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^n \subseteq \mathbb{R}^p \times \mathbb{R}^q$ scaled to the Cartesian product of the $p-$ and $q-$dimensional unit hypercubes, and that we have completed computer model runs $\eta(\mathbf{x}_i, \mathbf{t}_i)$ for $i = 1, \cdots, n$. Define the GP prior for modeling $\eta(\cdot, \cdot)$ as follows. Let the mean function $\mu(\mathbf{x}, \mathbf{t}) = c$, where c is a constant. Set the covariance function in terms of the marginal precision λ_η and a product power exponential correlation function:

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

where each β_k describes the strength of the GP's dependence on one of the elements of the input vectors $\mathbf{x}$ and $\mathbf{t}$, and α_η determines the smoothness of the GP. The model is completed by specifying priors for the hyperparameters $c, \lambda_\eta, \alpha_\eta$, and β_j^η for $j = 1, \ldots, p+q$, though in practice these are often set to predetermined values.

2.2 Design to target outcomes

Call design targets treated as observations in the design procedure we propose below "target outcomes", and call that procedure, which uses a Bayesian model calibration framework with target outcomes in place of real observations, "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. Just as in traditional calibration, in which the result of the procedure 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. The posterior predictive distribution is thereby pushed toward the target outcomes. The model used is assumed to yield accurate results within the design domain. Future work in this area will address the case of models that are known or suspected to suffer from systematic bias.

The tools of model calibration founded in the work of Kennedy and O'Hagan (2001) retain their advantages under our proposed methodology. Most centrally, calibration to target outcomes $\mathbf{y}$ produces not merely a point estimate $\mathbf{t}^*$, but rather a posterior distribution of $\mathbf{t}|\mathbf{y}$ reflective of remaining uncertainty about the appropriate value of $\mathbf{t}^*$. Such uncertainty may have its source in parameter uncertainty (uncertainty about the values of model inputs other than the design variables), model form uncertainty (uncertainty about how closely the code approximates reality), and that which traditional calibration would consider observation error. Of course, targets are not actually observations, so the concept of observation error does not cleanly transfer. However, a similar uncertainty would be uncertainty about which specific target values best reflect one's design goals. The Bayesian

model calibration framework allows for the quantification of all of these uncertainties. Furthermore, by the use of informative priors on the model discrepancy and observation error, the identifiability concerns of the Kennedy-O'Hagan approach can be mitigated (Bayarri et al., 2007; Tuo and Wu, 2016).

In general, target outcomes should aim only a little beyond what is realistically achievable; only as much as is necessary to ensure the targets are at least as ambitious as any true optimum in the system. Two reasons why one should go only a little beyond that are as follows: Firstly, if target outcomes are set to be too farfetched, then the procedure can become computationally unstable due to underflow and round-off error, since any value of $\boldsymbol{\theta}$ within its support will have extremely low likelihood. Secondly, increasing the distance of the target outcomes from the optimal region reduces the identifiability of that region. This is the same effect as the general case when observation error variance is much larger than the variance of the prior distribution on a parameter; i.e., the posterior is much more strongly determined by the prior than the likelihood, resulting in limited Bayesian learning about quantities of interest.

It is common to plug in the MLEs 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 a computer model emulator using the target outcomes, which do not arise from the model (see Liu et al., 2009, on the dangers of training an emulator on inappropriate data). Therefore, we use values found by maximizing the log likelihood of the observations of the simulation with

respect to λ_η and $\boldsymbol{\beta}^\eta$. We set the GP to have constant mean $c = 0$, which works well when (as here) the GP is not used for extrapolation (Bayarri et al., 2007). We set $\alpha_\eta = 2$, which assumes that the model output is infinitely differentiable.

We similarly model the discrepancy term $\delta(\cdot)$ as a GP, also with mean zero, and with covariance function $C_\delta(\mathbf{x}, \mathbf{x}') = \lambda_\delta^{-1} \prod_{k=1}^{p} \exp\left(-\beta_k^\delta |x_k - x_k'|^{\alpha_\delta}\right)$. This is included in the model to capture systematic discrepancy, not between the model and the true system (since we here assume our model is uniformly valid), but between target outcomes and the feasible design space. We use priors $\rho_k^\delta \sim \text{Beta}(1, 0.3)$, where $\rho_k^\delta = \exp(-\beta_k^\delta/4)$ for $k = 1, \ldots, p$. Details surrounding the choice of prior for λ_δ are discussed below. As with the covariance function of $\eta(\cdot, \cdot)$, we set $\alpha_\delta = 2$.

Let $\boldsymbol{\eta} = (\eta(\mathbf{x}_1, \mathbf{t}_1), \cdots, \eta(\mathbf{x}_n, \mathbf{t}_n))^T$ be the vector of completed runs of the simulator, $\mathbf{y} = (y(\mathbf{x}_{n+1}), \cdots, y(\mathbf{x}_{n+m}))^T$ the target outcomes we wish to induce the system to achieve, and $\mathcal{D} = (\boldsymbol{\eta}^T, \mathbf{y}^T)^T$. Then $\mathcal{D}|\boldsymbol{\theta}, \widehat{\lambda_\eta}, \widehat{\boldsymbol{\rho}^\eta}, \lambda_\delta, \boldsymbol{\rho}^\delta$ is distributed as multivariate normal with mean 0 and covariance $\mathbf{C}_\mathcal{D}$, a matrix with i, j entry equal to $C((\mathbf{x}_i, \mathbf{t}_i), (\mathbf{x}_j, \mathbf{t}_j)) + I(i, j > n) \cdot (C_{obs}(\mathbf{x}_i, \mathbf{x}_j) + C_\delta(\mathbf{x}_i, \mathbf{x}_j))$. $C_{obs}(\cdot, \cdot)$ serves as "observation error" of our own target outcomes, since in a typical case one can at best identify a small target region within which the choice of any particular point as a target outcome would be arbitrary. Thus, $C_{obs}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \delta_{ij}^K$, where δ^K is the Kronecker delta and σ^2 is chosen to reflect the desired tolerance level, such that targets within σ of each other are considered roughly equivalent. For our applications, we set $\sigma^2 = 0.05$ (with target outcomes standardized to have mean zero and prior standard deviation 1). Notice also that $C_{obs}(\cdot, \cdot)$ serves to add a nugget to the covariance matrix produced by $C_\delta(\cdot, \cdot)$. Aside from easing matters computationally (by

improving the conditioning of the covariance matrix), the addition of such a nugget can improve the quality of the fit of the GP discrepancy δ (Gramacy and Lee, 2012). Setting a uniform prior on the design variables $\boldsymbol{\theta}$, the joint posterior density under the model is

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

Markov chain Monte Carlo methods are used to explore the posterior distribution.

To successfully locate the optimal design region, one must either to place an informative prior on the marginal precision λ_δ of the discrepancy $\delta(\cdot)$, or else specify λ_δ outright. Otherwise, the optimal region of the design variable space will suffer from poor identifiability, which in our case will mean that the posterior distribution will not be concentrated within the optimal region. This longstanding concern was raised in the discussion of Kennedy and O'Hagan (2001), as well as by Bayarri et al. (2007), Tuo and Wu (2015), and Plumlee (2017). How informative one's prior on λ_δ will be depends upon how much one knows about the true Pareto front prior to undertaking CTO. For instance, if in a univariate case it is known with some confidence that the true optimum is nearly constant as a function of the other model inputs, and that it occurs in the interval $[10, 11]$, then a constant target outcome of 9 could be used with an informative prior tailored to this prior knowledge of the approximate resulting discrepancy.

Where the prior on λ_δ cannot be chosen to be *accurate* (due to insufficient prior knowledge of the Pareto front) it should be chosen to *overestimate* the precision. Otherwise, underestimation of λ_δ may lead to poor identifiability of the optimal design region by granting the model too much flexibility in systematically deviating from the targets. However, if λ_δ is too highly overestimated, then MCMC may become trapped in a local mode,

leading to convergence problems. In short, while the proposed methodology is forgiving of overestimation of λ_δ, the identifiability of the optimal design region(s) is best served by supplying as informative of a prior as possible.

In situations where one lacks the prior knowledge necessary to select target outcomes near the Pareto front and an accurate prior for λ_δ, an alternative is to use a "preliminary round" of CTO to *estimate* the Pareto front. For example, consider again the univariate case, supposing now that we know only that the optimal output is approximately constant somewhere in the range $(0, 20)$. One could perform CTO directly with constant target outcome below the known feasible design space (e.g., -1) and prior on λ_δ that weakly pulls the posterior predictive distribution toward the lower end of its range (e.g., exponential with rate 2). If the true optimal output turns out to be close to the lower end of its known range, this will likely work well. However, if the optimal output turns out to be sufficiently high, the optimal region of the parameter space could suffer from poor identifiability. To avoid this, one can instead perform CTO still with constant target outcome -1 but with a prior on λ_δ that deliberately exploits the identifiability problems of the Kennedy-O'Hagan framework in order to explore large regions of the parameter space – say, exponential with rate 0.1. Though the resulting posterior distribution will have greater density in the optimal region(s) than would a uniform sampling, it will likely not center in the optimal region, instead covering a larger area of the design space. The resulting posterior predictive distribution can be filtered to retain only its Pareto front, and this can be used as an estimate of the true Pareto front in the vicinity of the target outcome. The preliminary estimate allows one to select a new set of target outcomes that is known to lie

Algorithm 1: Full CTO procedure including preliminary estimation of Pareto front	
1.	Set target outcomes to lie outside of the feasible design space.
2.	Set a vague prior on λ_δ.
3.	Perform MCMC and use the resulting posterior samples of $\boldsymbol{\theta}$ to draw from the posterior predictive distribution of the model output.
4.	Filter the resulting predictions to retain only the Pareto optimal values. The remaining set of values $\mathcal{P}$ estimates the Pareto front.
5.	Select new target outcomes using $\mathcal{P}$. These may be entirely new targets chosen after examining $\mathcal{P}$, or may simply be updated automatically to set each target outcome to be the same (small) distance $1/\sqrt{\lambda}$ (for some λ) from $\mathcal{P}$.
6.	Set a strong (or degenerate) prior on λ_δ with mean λ.
7.	Perform MCMC in the usual way.

near the optimal region, along with an accurate and informative prior on λ_δ that reflects the estimated distance between the new target and the optimal region. Performing CTO with these new targets and prior will result in a posterior distribution that concentrates on the optimal region, and the resulting posterior predictive distribution will allow one to estimate the optimal output with appropriate quantification of the uncertainty. The full CTO process, including preliminary estimation of the Pareto front, is given in Algorithm 1.

An illustration of the benefits of preliminary CTO appears in Figure 1. Suppose that, prior to undertaking CTO, we know only that the model outputs are positive. Then $(0, 0)$ is a natural choice to use as a target outcome. However, in this case, that choice of target outcome will yield three problems. Firstly, the feasible design space is distant from $(0, 0)$. As a result, the optimal region is, relative to the size of the design space, not much closer to the target outcome than other regions of the design space. Indeed, the farthest point in the entire design space is only 2 times farther from $(0, 0)$ than is the nearest point. This leads to very poor identifiability of the optimal region, since large portions of the design space are roughly as close to $(0, 0)$ as is the optimum. Secondly, the optimal region

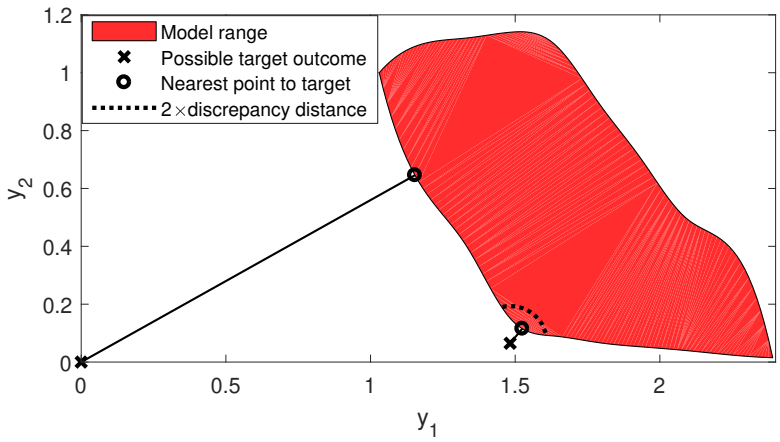

Figure 1: Two potential choices of target outcomes. The distance from $(0, 0)$ to the farthest point in the feasible design space is only 2 times the distance to the nearest point. By contrast, for the point (1.32,0.065), the dotted line shows the small region of the design space within 2 times the distance to the nearest point.

determined by the choice of $(0, 0)$ is somewhat arbitrary. Notice that in this example the entire "left" and "bottom" edges of the design space comprise the Pareto front. The point closest to $(0, 0)$ is not intrinsically superior to any other point on the Pareto front. Its uniqueness lies 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. Thirdly, since we don't know the range of the feasible design space, we are not in a position to set an informative prior for λ_δ. By contrast, suppose now that we have performed preliminary CTO and have available a rough estimate of the Pareto front, empowering us to choose another point as our target outcome – for example, the point $(1.32, 0.065)$ pictured in Figure 1 targets a point of diminishing returns where allowing y_1 to increase further leads to diminishing returns in reducing y_2. Alternatively, one could select $(1, 0.015)$, since this is the minimum of each

model output, and thus will optimize to a region minimizing the sum of squared increases in each output over its minimum. Either choice of new target outcome would answer all of the above problems. Firstly, each point is much closer to the design space, leading to greater identifiability. Secondly, these choices of target outcome and resulting optima are not arbitrary, but rather are driven by articulable goals informed by an estimate of the Pareto front. Thirdly, with a rough estimate of the Pareto front we can supply a strong (or even degenerate) prior for λ_δ. Note also that when an emulator is used, a preliminary round of CTO can use the same set of model observations as the subsequent CTO for the training points of the emulator. So performing preliminary CTO does not add to the total budget of model runs, and can thus be a computationally cheap supplement to CTO.

3 Simulated Example

As an illustration of our proposed procedure with known solution, consider the following problem of minimizing a function with trivariate output. Let $(x, \boldsymbol{\theta})$ be the vector of inputs, with scalar control input $x \in [1.95, 2.05]$ and design variables $\boldsymbol{\theta} = (\theta_1, \theta_2) \in [0, 3] \times [0, 6]$. We seek optimal settings for $\boldsymbol{\theta}$. We consider three outputs: $y_1 = \left(\theta_1 \exp\left(-\left(\theta_1 + |\theta_2 - \frac{\pi x}{2}|\right)\right) + 1\right)^{-1}$, $y_2 = \left(\theta_2^{x-1} \exp\left(-0.75\theta_2\right) + 1\right)^{-1}$, and $y_3 = 15 + 2\theta_1 + \theta_2^2/4$. We suppose that prior to undertaking CTO we know only that the model outputs are positive. Figure 2 displays the y_1, y_2, and y_3 surfaces as functions of θ_1 and θ_2 at $x = 2$. Assuming an easily evaluated model (so that a GP code surrogate is not needed), we have $\mathbf{z}(x) = \mathbf{f}(x, \boldsymbol{\theta}) + \delta(x) + \boldsymbol{\epsilon}$ for target outcome $\mathbf{z}$, where $\mathbf{f} = (y_1, y_2, y_3)^T$ is the model output, $\delta(\cdot)$ is the discrepancy between the optimal region and the target outcome, and $\boldsymbol{\epsilon}$ is the target outcome tolerance and follows

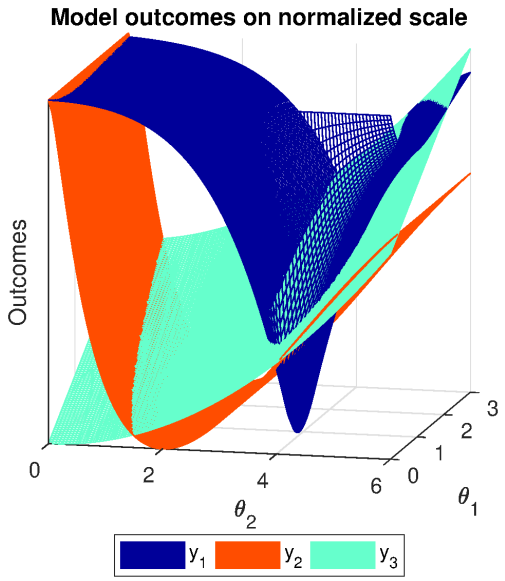

Figure 2: True outputs of the example model.

a $N(\mathbf{0}, 0.05I)$ distribution.

We initially set the target outcomes to $(0, 0, 0)$, constant as a function of x. We then estimated the Pareto front via a preliminary round of CTO with $\lambda_\delta \sim \text{Exp}(1)$ in order to estimate the standardized distance of the target outcome from the Pareto front. We filtered the resulting posterior predictive distribution to retain only the Pareto optimal points. Rescaling so that each model output y is replaced with $y^* = (y - \mathbb{E}(y))/\sqrt{\mathbb{V}(y)}$, the distance from the Pareto front to the target outcome is 16 units. This is large compared to the size of the design space, at roughly four times its diameter. As a result, the use of $(0, 0, 0)$ as a target outcome would lead to poor identifiability of the optimal region. To improve identifiability of the optimal region, we updated the target outcome to lie along the line connecting the original target outcome to the nearest point of the estimated Pareto front, but now closer to the latter. We chose a distance of one unit away, as this is roughly the sample standard deviation of the distances from each model output to the sample

mean. We thereby approach the estimated Pareto front closely (relative to the size of the design space) while remaining confident that the new target outcome of $(0.71, 0.71, 17.92)$ still outperforms the true Pareto front. This confidence is based on the observation that $F(\mathbf{x} - (0.71, 0.71, 17.92)^T) > 0.95$, where x is the nearest point of the estimated Pareto front and F is the cdf of the tolerance $\boldsymbol{\epsilon} \sim N(\mathbf{0}, 0.05I)$. We then set the marginal precision of the discrepancy function $\lambda_\delta = 1$ for subsequent CTO, corresponding to a degenerate prior informed by the estimated distance of the new target outcome from the Pareto front. For comparison, we also performed CTO directly, without the use of preliminary CTO. To do so, we used our original target outcome of $(0, 0, 0)$ with a Gamma(10,10) prior deliberately overestimating λ_δ. Notice that although the Gamma prior from direct CTO and the degenerate prior $\lambda_\delta = 1$ from full CTO have the same means, this is an overestimation only in the case of direct CTO, since the two posterior explorations used different target outcomes. Figure 3 shows the resulting posterior distributions of the two design procedures, including the marginal distributions of the design variables. 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 of the two cases, and peaks near the true optimum. However, the benefits of preliminary CTO are apparent in the greater spread of the posterior distribution from direct CTO. The marginals are much more sharply peaked after using preliminary CTO, with much lighter tails. Thus relying on an estimate of the Pareto front when selecting target outcomes for CTO can greatly reduce uncertainty about the optimal settings for $\boldsymbol{\theta}$ and for the resulting performance and cost outcomes for the system. This simulation example illustrates that

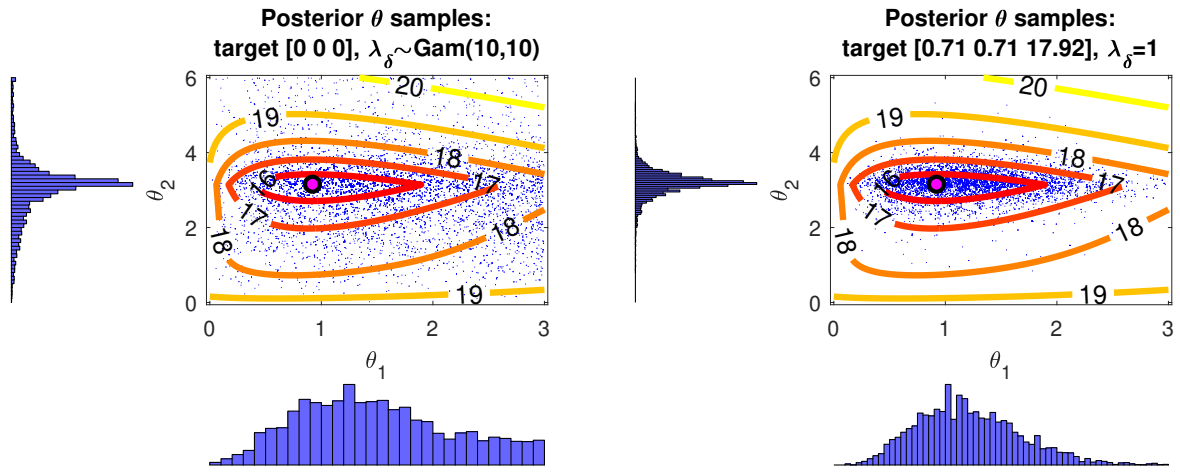

Figure 3: Posterior draws from CTO in the simulated example without and with the use of preliminary CTO. The contours show, for each point in the design space, the Euclidean distance of the model output at that point from the original target outcome (0,0,0), averaged across the control input range $[1.95, 2.05]$. The large dot shows the true optimum.

CTO can be used directly with little foreknowledge of a system's Pareto front, but that greater identifiability of the optimal region can be achieved using preliminary CTO.

4 Application

In this section we apply our proposed approach toward designing a material to be used in a wind turbine blade of fixed geometry. The goal is to wed the typically separate tasks of material selection and material design, thereby designing a composite material to optimize performance in a turbine blade. Our computer model here uses `ANSYS` finite element analysis software (ANSYS, Inc., 2017). It is important to observe that we assume the finite element model is an accurate representation of reality.

4.1 Wind turbine blade design

Two primary performance targets for the design and construction of wind turbine blades are the distance (in meters) that the blade tip deflects under load from its starting position, and the angle of twist the blade experiences under load. Here, the goal of engineering design is to keep these measures and the material cost as small as possible. The blade is to be a composite of two given materials, one serving as the matrix and the other the filler. For the wind turbine blade, given a fixed choice of matrix and filler, the properties of the composite depend on the volume fraction (the volume ratio of filler material to matrix material used in the composite) and the thickness of the shear web used in the blade. The resulting material properties impact the performance of the blade, as well as its cost per square meter. Temperature affects the composite material properties and hence the blade tip deflection and angle of twist. The finite element model takes as inputs a triplet (h, v, k), where h is the operating temperature of the wind turbine (in kelvin), v is the volume fraction of the material, and k is the thickness of the material (in mm). The output of the model is the triplet (d, r, c), where d is tip deflection (in meters), r is twist (in radians), and c is cost per square meter (USD) to construct the material. The wind turbine is deemed to operate over the range of temperatures 230K-330K.

4.2 Emulation of finite element model

The finite element simulator is too computationally expensive to be suitable for direct use in an MCMC routine. We employed a GP emulator in the manner of Williams et al. (2006). For this purpose, we drew 500 (trivariate) observations from the finite element

simulator. The inputs were determined by a Latin hypercube sampling design (McKay et al., 1979) based on plausible ranges for the three inputs as identified by subject matter experts: $[230\text{K}, 330\text{K}] \times [.2, .6] \times [10\text{mm}, 25\text{mm}]$. We took the computer output to follow a GP with mean 0 and product power exponential covariance function as given in (2). Since the model output is trivariate, we use two binary dummy variables a_1, a_2 to convert the model to univariate output with five inputs, in order to employ a univariate GP emulator. Thus for given values of h, v, k: setting $a_1 = a_2 = 0$ outputs the tip deflection d, setting $a_1 = 1$ and $a_2 = 0$ outputs the twist r, and settings $a_1 = 0$ and $a_2 = 1$ outputs the cost c.

The hyperparameters $\lambda_\eta, \boldsymbol{\beta}^\eta$ are estimated to plug in to the prior via maximum likelihood using only the finite element model output. We used `fmincon()` (MATLAB, 2017) to maximize (with $\mathcal{D} = \boldsymbol{\eta}$) over the joint (six-dimensional) support of $\boldsymbol{\beta}^\eta, \lambda_\eta$. The result is that, following the form of Equation (2) with $p = 3$, $q = 2$, and $(x_1, x_2, x_3, t_1, t_2) = (a_1, a_2, h, v, k)$, we have $\hat{\lambda}_\eta = 0.0152$ and $\hat{\boldsymbol{\rho}}^\eta = (0.9358, 0.6509, 0.6736, 0.4797, 0.9673)$, where $\rho_k^\eta = \exp(-\beta_k^\eta/4)$.

4.3 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 simulation responses has mean 0 and standard deviation 1. The full joint posterior density of the design variables and discrepancy function hyperparameters is given in Equation (3), using the MLEs given above.

The initial target outcomes were set to $(0, 0, 0)$ on the original scale, constant as a function of temperature, on an evenly-spaced grid of temperature values over the range

[230K, 330K]. We carried out a preliminary round of CTO with an Exp(5) prior on λ_δ, in order to estimate the Pareto front and update the target outcomes to lie close to the Pareto front and thereby improve identifiability of the optimal region. For this purpose, a total of 2,000 Markov chain realizations 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 1,000 draws were discarded as burn-in. During the burn-in period, the covariances of the proposal distributions were periodically adjusted for optimal acceptance rates of around 23% for the multivariate $\boldsymbol{\theta}$ and $\boldsymbol{\rho}^\delta$ (Roberts et al., 1997) and 44% for the scalar λ_δ (Gelman et al., 2013, p. 296) using the sample covariance of the preceding draws. Convergence of the three chains was verified visually and by the Gelman-Rubin statistic (≈ 1.01; Gelman and Rubin, 1992). As expected for the preliminary round of CTO, the posterior distribution of $\boldsymbol{\theta}$ 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 design space is three-dimensional, the Pareto front appears to be a roughly coplanar curve describing a trade-off in which reduced cost is achieved by allowing higher deflection and twist. We see a distinct point of maximum curvature in the Pareto front. This location seems to represent a point of diminishing returns in the tradeoff between performance and cost, and thus we selected this point as the target region for design. To do so, we set the point (deflection $= 0.75$m, twist $= 0.09$ rad, cost $= \$130.34$) as the target outcome, constant as a function of temperature. Based on the estimated Pareto front, the target outcome is approximately 0.2 units away on the standardized scale. Therefore, we

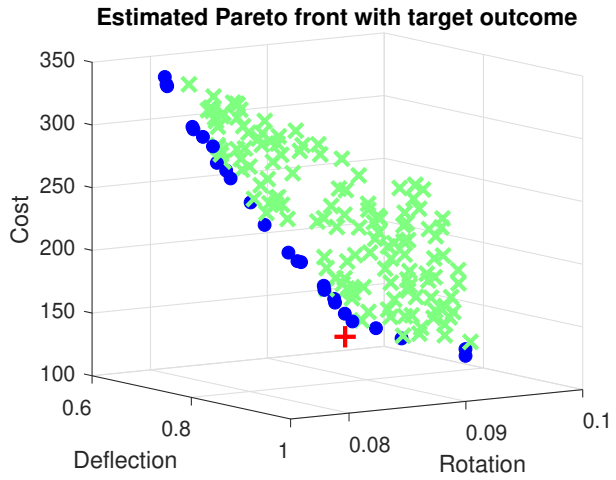

Figure 4: Each x represents a dominated design drawn from the predictive distribution through preliminary CTO in the wind turbine design application. The dots indicate the estimated Pareto front. The plus sign is the target selected as the performance objective in our proposed design approach.

set $\lambda_\delta = 1/0.2^2 = 25$.

In the subsequent CTO step, we employed the same MCMC approach as in the preliminary round, except that λ_δ was now fixed. The marginal posterior distribution of the design variables is shown in Figure 5 as contours of highest density regions. The contrast of the posterior distribution with the prior, which is uniform over the area shown in the figure, indicates that strong Bayesian learning has occurred in the design process. The prior and posterior marginal predictive distributions of the model outputs are shown in Figure 6, where the prior predictive distributions are based on a uniform sampling of the model inputs over their domains. Each marginal posterior density peaks sharply near the target outcome. The mean model output under the prior is (deflection $= 0.76$m, twist $= 0.09$ rads, cost $=$ \$207.90/m^2), whereas under the posterior it is $(0.76\text{m}, 0.09\text{ rad}, \$149.47/\text{m}^2)$. Though the

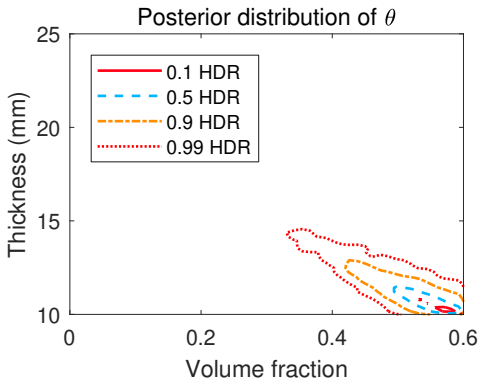

Figure 5: Contours of highest density regions of the posterior distribution from CTO in the wind turbine blade system. The prior is uniform over the area shown.

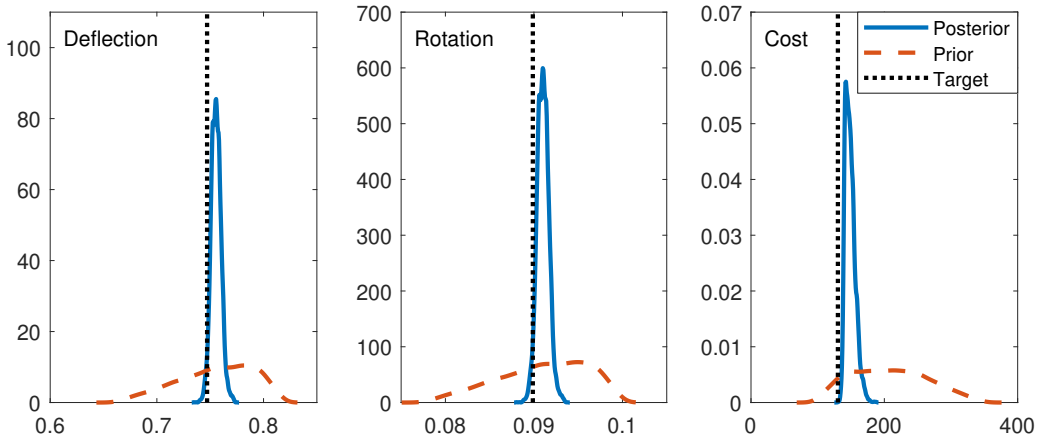

Figure 6: Approximate prior and posterior marginal predictive densities for each of the three outputs. Note that it is to be expected that the posteriors peak near the target (and not on it), since the target was intentionally chosen to lie outside the feasible design space.

mean performance outcomes are approximately the same under the posterior distribution as under the prior, the mean cost per square meter and the uncertainty of the outcomes have both been dramatically reduced. If one desires instead to prioritize gains in performance over cost, this can be accomplished by selecting target outcomes that reflect those priorities, as we discuss in Section 4.4.

4.4 Pareto front estimation with 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 $d-$dimensional, one may draw a grid over the range of $d-1$ of the model outputs and perform CTO to minimize the remaining output at each point of the grid. The $d-1$ outputs, at each grid point, are treated as known up to observation error (meaning that the discrepancy function $\delta(\cdot)$ is set to 0 in the dimension of these outputs). 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 simplicity, twist has been removed as a model output, leaving a system with 2-dimensional

output of deflection and cost. The range of cost is known (via preliminary CTO) to be [\$96, \$352]. A 20-point grid was drawn over this range of costs. For each point c in the cost grid, we used the point (0m, \$$c$) as the target outcome for calibration (constant with respect to temperature). For each such point, we then updated this initial target outcome to improve identifiability using the rough estimate of the Pareto front from preliminary CTO using target outcome (0m, \$0). Note that only one round of preliminary CTO was needed for this purpose, rather than a separate instance at each grid point.

The result of the strategy is to provide an estimate of the response surface with included uncertainty quantification describing, for each point in the grid, the minimal achievable outcome for the output not included in the grid. This enables a decision-maker to visualize the space of desirable possibilities with associated uncertainties. The result of applying this strategy to the wind turbine blade application is shown in Figure 7. The lefthand plot shows that the posterior model predictions respected the target cost values used in the design process. The Pareto front for the system, along with associated uncertainty bands, appears in the righthand plot.

The use of CTO in the wind turbine calibration case illustrates how preliminary CTO may be used, not merely to improve the identifiability of a pre-determined optimal region as in Section 3, but rather to identify a desirable region of the design space and select target outcomes that design to that region. In the wind turbine case, selecting $(0, 0, 0)$ as one's target determines the optimal region to be the high-cost region toward the upper-left of Figure 4, since (on the standardized scale of model outputs) that region happens to be closest to the target. If one has substantive goals that drive one to select that target, then

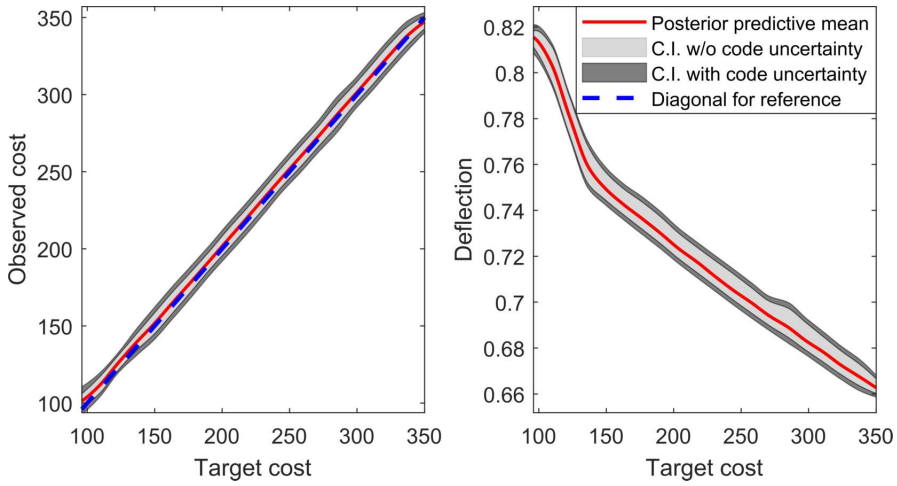

Figure 7: The lefthand plot shows the posterior predictive distribution of cost as a function of target cost, verifying that the calibration achieved the target costs up to small error. The righthand plot shows the posterior predictive distribution of minimum feasible deflection as a function of target cost, and therefore is an estimate of the Pareto front for the system with attendant uncertainty quantification.

one is well-served by optimizing to that high-cost region. But if the target $(0, 0, 0)$ is chosen arbitrarily, then the resulting optimal region is itself determined arbitrarily. The estimate of the Pareto front provided by preliminary CTO allows us to identify regions of special interest, and to select target outcomes that that lead to clearly defined designs, as illustrated in Figure 4. The use of CTO in this case also 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 (.58, 10.2mm), but also that if the volume fraction is lowered to 0.4 it is important to simultaneously raise the thickness to 14mm. More generally, one can see in Figure 5 an indication of the range of $\boldsymbol{\theta}$ values that achieve results near the target, which is potentially useful when one's goal is

to set tolerances (rather than a specific value) for $\boldsymbol{\theta}$. Finally, the use of CTO in the wind turbine case illustrates how the 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.

5 Conclusion

We have described how the computer model calibration framework established by Kennedy and O'Hagan (2001) can be adapted to address questions of engineering design. Calibration to target outcomes is a modification of that framework which undertakes design by "calibrating" a computer model not to field observations, but rather to artificial data representing performance and cost targets for the system. 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 is induced to approximate the target outcomes, so that the posterior distribution of $\boldsymbol{\theta}$ constitutes a distribution on optimal design settings for the system. Repeated applications of this methodology allow one to construct a thorough estimate of the Pareto front of the system with quantified uncertainties. Unlike other methods of Bayesian optimization (a review of which was provided by Shahriari et al. 2016), CTO does not require the ability to evaluate model output adaptively. Instead, it can operate using 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. We have shown thereby a variety of ways in which CTO can be used to guide decision-makers in the design process. By expropriating established tools of model calibration, CTO offers a method of optimization which is sensitive to, and quantifies, all sources of uncertainty.

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 second discrepancy term capturing model bias. That is, while CTO as presented above includes $\delta(\cdot)$ to capture systematic discrepancy between the target outcomes and the true system, it assumes that the computer model is an unbiased estimate of the true system. The inclusion of a second discrepancy term to capture systematic differences between the model and the observed reality would avoid this idealizing assumption. 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 control inputs.

SUPPLEMENTARY MATERIAL

Matlab code for CTO: This includes the example model described in Section 3, along with code to perform CTO on that system and thereby reproduce Figure 3.

References

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.

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).

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, nov). 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.

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.

Gramacy, R. B. and H. K. H. Lee (2012). Cases for the nugget in modeling computer experiments. *Statistics and Computing 22*(3), 713–722.

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

Hemez, F. M. and S. Atamturktur (2011, sep). 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.

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.

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.

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.

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

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

Plumlee, M. (2017). Bayesian calibration of inexact computer models. *Journal of the American Statistical Association 112*(519), 1274–1285.

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.

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.

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.

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 C. F. J. Wu (2015). Efficient calibration for imperfect computer models. *Annals of Statistics 43*(6).

Tuo, R. and C. F. J. Wu (2016). A theoretical framework for calibration in computer 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, feb). 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, jul). Simulating the dynamics of wind turbine blades: partII, model validation and uncertainty quantification. *Wind Energy 16*(5), 741–758.

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.