Combining model calibration and design

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,
Christopher Kitchens
Department of Chemical and Biomolecular Engineering, Clemson University,
and
Sez Atamturktur
Department of Architectural Engineering, Pennsylvania State University

Keywords:

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

1 Introduction

The goal of traditional Kennedy-O'Hagan style calibration (KOH, Kennedy and O'Hagan, 2001) is to find a posterior distribution on unknown parameters by calibrating a computer model using real-world observations of the modeled phenomenon. By contrast, the design methodology of calibration to target outcomes (CTO) uses the KOH framework to find a posterior distribution on optimal input settings in the model by "calibrating" a computer model using artificial observations that reflect performance and cost targets for the modeled system. The goal of the work described here is to combine KOH and CTO. Call the resulting methodology DCTO, for dual calibration to target outcomes.

CTO as previously developed assumes, somewhat idealistically, that the computer model is already perfectly calibrated. DCTO avoids this idealization. Furthermore, when undertaking KOH, some areas of the model range may be of greater interest than others. For example, one may be more interested in calibrating the model to be accurate in the optimal region of some design variable θ than elsewhere. Undertaking dual calibration may allow us to focus our calibration efforts on such regions of interest, prioritizing them over other areas of the model range.

2 Naïve DCTO

The version of KOH considered here is that which finds a posterior distribution on a parameter of interest, $\boldsymbol{\theta}$, using a GP emulator with hyperparameters $\widehat{\boldsymbol{\phi}_\eta}$ estimated via maximum likelihood using a budget of computer model observations $\boldsymbol{\eta}$. Similarly, we use

a GP prior with hyperparameters ϕ_δ to model discrepancy between the computer model $\eta()$ and the true function $f()$ that it represents. In the work described here, we employ stationary GPs with a Gaussian kernel covariance structure $C(\mathbf{x}, \mathbf{x}') = 1/\lambda \times \exp(-\beta(\mathbf{x} - \mathbf{x}')^2)$, so that $\widehat{\boldsymbol{\phi}_\eta} = \left[\widehat{\beta},\ \widehat{\lambda}\right]$. Setting priors on $\boldsymbol{\theta}$ and on ϕ_δ, we train the GP emulator on observations $\boldsymbol{\eta}$ and use MCMC to explore the distribution

$$\pi(\boldsymbol{\theta}, \phi_\delta | \mathcal{D}, \widehat{\phi_\eta}) \propto \pi(\mathcal{D} | \boldsymbol{\theta}, \widehat{\phi_\eta}, \phi_\delta) \times \pi(\boldsymbol{\theta}) \times \pi(\phi_\delta) \tag{1}$$

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

In a computer calibration problem, $\mathbf{y}$ is a set of observations of the system modeled by $\eta()$. In CTO, by contrast, y is a set of target outcomes – artificial data representing the way that one wishes to induce the system to behave, rather than observations one has made of the system in reality. When one wishes to perform CTO on a system that also requires traditional calibration, then, one obvious idea is to combine the two approaches by using Equation (1) with $\mathbf{y} = (\mathbf{y}_r^T,\ \mathbf{y}_t^T)^T$, an array containing both real observations $\mathbf{y}_r$ (for calibration) and target outcomes $\mathbf{y}_t$ (for CTO). However, this approach will not work, for two reasons. Firstly, the inputs $\boldsymbol{\theta}$ are typically not the same in calibration and in CTO. In calibration, one seeks to estimate the value of an input that either represents some true unknown quantity, or else which induces the model to represent some true unknown quantity. In CTO, one seeks to find a distribution on an input that is under the researcher's control.

Relatedly, by including target outcomes in one's observations $\mathbf{y}$ used in KOH calibration, one compromises the integrity of the calibration. KOH calibration, after all, aims to use

one's observations of the real system to estimate the value of $\boldsymbol{\theta}$. If one's observations $\mathbf{y}$ do not reflect the behavior of the real system, then they will constitute a poor source of information for bringing the model into alignment with reality. Simply put, to represent reality one must train one's model on observations of reality, not on unobserved targets.

3 Dual calibration to target outcomes

Given that naïve DCTO cannot accomplish the dual tasks of calibration and design, a different approach must be taken. The two tasks must be separated. An obvious choice here is to perform KOH calibration first, without involving any target outcomes, and then to use the calibrated model in order to perform CTO. Under this approach, with observations $\mathbf{y}_r$ of the system of interest, one would employ the model described in Equation (1) with $\boldsymbol{\theta} = \boldsymbol{\theta}_c$ (the parameters to be calibrated) and with $\mathcal{D} = \mathcal{D}_c = (\boldsymbol{\eta}^T, \mathbf{y}_c^T)^T$. The result would be a posterior distribution of $\boldsymbol{\theta}_c$ and of $\delta(\cdot)$, the systematic discrepancy between the computer model $\eta(\cdot, \cdot)$ and the true system $f(\cdot)$. These can be used to produce estimates $\widehat{\boldsymbol{\theta}}_c$ and $\widehat{\delta}(\cdot)$ such that $f(\mathbf{z}) \approx \eta(\mathbf{z}, \widehat{\boldsymbol{\theta}}_c) + \widehat{\delta}(\mathbf{z})$ for all $\mathbf{z}$ in the domain of f. The result is a calibrated model $\eta_c(\mathbf{z}) = \eta(\mathbf{z}, \widehat{\boldsymbol{\theta}}_c) + \widehat{\delta}(\mathbf{z})$ which can be used for CTO.

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

attempt to model any systematic discrepancy between η_c and f, since an estimate of that discrepancy is already included in η_c. For the purposes of Equation (1), this amounts to setting a degenerate prior on ϕ_δ at 0.

Above, the use of CTO following KOH relies on two separate implementations of Equation (1). It will be useful to produce an integrated model which describes the use of both procedures, and which makes clear the relationship between them. For this purpose, consider η as having three inputs $(\mathbf{x}, \mathbf{t}_c, \mathbf{t}_d)$ where $\mathbf{t}_c$ denotes the parameters targeted for KOH calibration, $\mathbf{t}_d$ denotes the input settings targeted for design via CTO, and $\mathbf{x}$ denotes the remaining known and/or controllable inputs. If η can be run quickly, then we use it directly in MCMC. However, if it is computationally expensive, we employ a surrogate by setting a Gaussian process (GP) prior on η with mean $m_\eta(\mathbf{x}, \mathbf{t}_c, \mathbf{t}_d)$ and covariance function $C_0((\mathbf{x}, \mathbf{t}_c, \mathbf{t}_d), (\mathbf{x}', \mathbf{t}_c', \mathbf{t}_d'))$. From here on in this discussion, assume that a GP surrogate is used for η. Model the systematic discrepancy between η and f at the true value of $\mathbf{t}_c = \boldsymbol{\theta}_c$ with another GP prior $\delta(\cdot, \cdot)$ having mean $m_\delta(\mathbf{x}, \mathbf{t}_d)$ and covariance function $C_\delta((\mathbf{x}, \mathbf{t}_d), (\mathbf{x}', \mathbf{t}_d'))$. In addition to systematic discrepancy between η and reality, measurement error ϵ_r may be included in the model for real observations $\mathbf{y}_c$, and additional Gaussian observation error ϵ_d may be included for target outcomes $\mathbf{y}_d$. The purpose of additional observation error ϵ_d is twofold. Depending on the distribution of ϵ_c, the target outcomes $\mathbf{y}_d$ may or may not be within the support of a model that lacks ϵ_d. Including ϵ_d ensures that the targets are compatible with the model. Secondly, including ϵ_d and estimating its variance σ_d^2 provides computational benefits. For example, even if the target outcomes are compatible with a model that does not include ϵ_d, they may be extreme

outliers to the extent that the relevant likelihoods are small enough to generate significant numerical errors during MCMC. In terms of the interpretation of the model, adding ϵ_d amounts to supposing that the observations made in ω were subject to greater than usual observation error, where that additional error is distributed as $N(0, \sigma_d^2)$. Though it is not necessary to assume that ϵ_c is Gaussian, for simplicity of presentation we assume here that it is distributed as $N(0, \sigma_c^2)$. Finally, we assume that η, δ, ϵ_c and ϵ_d are all mutually independent.

A collection of simulation runs is needed to train the GP code surrogate. Let $(\mathbf{x_s}, \mathbf{t_{cs}}, \mathbf{t_{ds}})$ be the design matrix for these simulation runs, and let $\mathbf{y_s}$ denote the output of these runs. Similarly, let $\mathbf{y_c}$ be observations made at $\mathbf{x_c}, \mathbf{t_c}$, and let $\mathbf{y_d}$ be target outcomes "observed" at $\mathbf{x_d}$. Finally, let $\mathbf{y} = (\mathbf{y_s}^T, \mathbf{y_c}^T, \mathbf{y_d}^T)^T$. Then it follows that $\mathbf{y} \sim \mathrm{N}(\mathbf{m}, \mathbf{C})$, where

$$\mathbf{m} = \begin{pmatrix} m_s(\mathbf{x_s}, \mathbf{t_{cs}}, \mathbf{t_{ds}}) \\ m_s(\mathbf{x_c}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t_d}) + m_\delta(\mathbf{x_c}, \mathbf{t_d}) \\ m_s(\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T) + m_\delta(\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_c^T) \end{pmatrix},$$

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

$$\mathbf{C_{11}} = C_0\left((\mathbf{x_s}, \mathbf{t_{cs}}, \mathbf{t_{ds}}), (\mathbf{x_s}, \mathbf{t_{cs}}, \mathbf{t_{ds}})\right)$$

$$\mathbf{C_{21}} = C_0\left((\mathbf{x_s}, \mathbf{t_{cs}}, \mathbf{t_{ds}}), (\mathbf{x_c}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t_{dc}})\right)$$

$$\mathbf{C_{31}} = C_0\left((\mathbf{x_s}, \mathbf{t_{cs}}, \mathbf{t_{ds}}), (\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T)\right)$$

$$\mathbf{C_{12}} = \mathbf{C_{21}}^T$$

$$\mathbf{C_{22}} = C_\eta\left((\mathbf{x_c}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t_{dc}}), (\mathbf{x_r}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t_{dc}})\right) + C_\delta\left((\mathbf{x_c}, \mathbf{t_{dc}}), (\mathbf{x_c}, \mathbf{t_{dc}})\right) + \sigma_c^2\mathbf{I}$$

$$\mathbf{C_{32}} = C_\eta\left((\mathbf{x_c}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{t_{dc}}), (\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T)\right) + C_\delta\left((\mathbf{x_c}, \mathbf{t_{dc}}), (\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_d^T)\right)$$

$$\mathbf{C_{13}} = \mathbf{C_{31}}^T$$

$$\mathbf{C_{23}} = \mathbf{C_{32}}^T$$

$$\mathbf{C_{33}} = C_\eta\left((\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T), (\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_c^T, \mathbf{1}\boldsymbol{\theta}_d^T)\right) + C_\delta\left((\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_d^T), (\mathbf{x_d}, \mathbf{1}\boldsymbol{\theta}_d^T)\right) + \sigma_c^2\mathbf{I} + \sigma_d^2\mathbf{I}$$

Note that when $\mathbf{y_d}$ and $\mathbf{x_d}$ are empty and $\mathbf{m}, \mathbf{C}$ reduce respectively to their first two and upper two-by-two block elements, this is simply the KOH framework. Thus, this combination of KOH and CTO generalizes the KOH framework.

This inclusive framework allows for KOH and CTO to be undertaken simultaneously. Call this inclusive framework DCTO, for dual calibration with target outcomes. A primary benefit of DCTO is that under this combined approach, the results of CTO include quantification of all sources of uncertainty. By performing KOH and then subsequently undertaking CTO using static estimates $\widehat{\boldsymbol{\theta}_c}$ and $\widehat{\delta}$ from KOH, uncertainty surrounding those estimates is not included in the results of CTO. Another benefit of the combined approach appears in cases in which $\boldsymbol{\theta}_c$ is suspected to be a function of $\boldsymbol{\theta}_d$. In such cases, one may be interested only or primarily in the value of $\boldsymbol{\theta}_c$ at the optimal value of $\boldsymbol{\theta}_d$. If one has

the freedom to sample adaptively from the true system, then this freedom can be applied in DCTO to concentrate samples disproportionately in the region of interest. This idea is explored further in Section XXX.

Another crucial difference between performing CTO after KOH and performing DCTO is the role of the targets $\mathbf{y}_d$ in the calibration of $\boldsymbol{\theta}_c$. In DCTO as described above, the likelihood of $\boldsymbol{\theta}_c$ is affected by $\mathbf{y}_d$, whereas in KOH $\mathbf{y}_d$ is not included in the model and hence cannot affect the distribution of $\boldsymbol{\theta}_c$. This is a point in favor of performing KOH separate from CTO, as $\mathbf{y}_d$ is artificial data and cannot plausibly serve as a source of information about the correct value of $\boldsymbol{\theta}_c$. A similar issue affects CTO when an emulator is used. In KOH, the hyperparameters of an emulator may be estimated prior to calibration (e.g. via maximum likelihood), or else one may set priors for these hyperparameters and sample from their posteriors during the calibration process. If the latter route is chosen for CTO, then one would again face a situation in which learning about real quantities from artificial data. The hyperparameters of the GP emulating η should be estimated using observations of η and of the real process that η simulates. The solution in the case of CTO is to employ some form of modularity (Liu et al., 2009). A modular analysis intentionally falls short of being a full Bayesian analysis, either for computational benefits, or to quarantine "suspect" aspects of the model. The target outcomes $\mathbf{y}_d$ are precisely such a suspect source of Bayesian learning—they are by their nature extreme outliers, and hence are a poor guide both for estimating the hyperparameters of the GP emulator and for estimating the parameter $\boldsymbol{\theta}_c$. In CTO, modularization with respect to the GP hyperparameters typically takes the form of producing maximum likelihood estimates of the GP hyperparameters and using those in

lieu of setting priors and exploring a posterior distribution. Modularization with respect to $\boldsymbol{\theta}_c$ is implicit in the fact that an estimate $\widehat{\boldsymbol{\theta}_c}$ is used in CTO after KOH. DCTO can similarly modularize simply by refraining from including $\mathbf{y}_d$ in the updates of $\boldsymbol{\theta}_c$ during MCMC. That is, rather than calculating the likelihood of a proposed sample $t_c^{(i+1)}$ at step i of the MCMC using $\mathbf{y} = (\mathbf{y}_s^T, \mathbf{y}_c^T, \mathbf{y}_d^T)^T$, one can instead calculate its likelihood using only $\mathbf{y}_r = (\mathbf{y}_s^T, \mathbf{y}_c^T) \sim N(\mathbf{m}_r, \mathbf{C}_r)$, where $\mathbf{m}_r$ and $\mathbf{C}_r$ are respectively the upper two and upper-left two-by-two components of $\mathbf{m}$ and $\mathbf{C}$. Such modularization ensures that all Bayesian learning of $\boldsymbol{\theta}_c$ is based upon the real observations rather than upon $\mathbf{y}_d$.

4 Example with simulated data

Consider the function of three inputs $\eta(x, t_c, t_d) = x/(t_d^{t_c - 1} \exp(-0.75 t_d) + 1)$. Figure 1 shows the output of this function for $x = 1$ over the range $(t_c, t_d) \in [1.5, 4.5] \times [0, 5]$. We arbitrarily set $\theta_c = 2$ to be the "true" value of t_c. For any value of x and t_c, the optimal (minimizing) value of t_d is $(4/3)(t_c - 1)$, so we have $\theta_d = 4/3$. Figure 2 shows the locations of the true and optimal values (respectively) of θ_c and θ_d. There it is clear that the true value of θ_c is far from optimal – if this value were within our control, its optimal value would be at the upper end of its support, at 4.5. Thus η showcases the ability of DCTO to perform simultaneously both calibration and design in the case when our "truth-seeking" goals and our design goals are in tension.

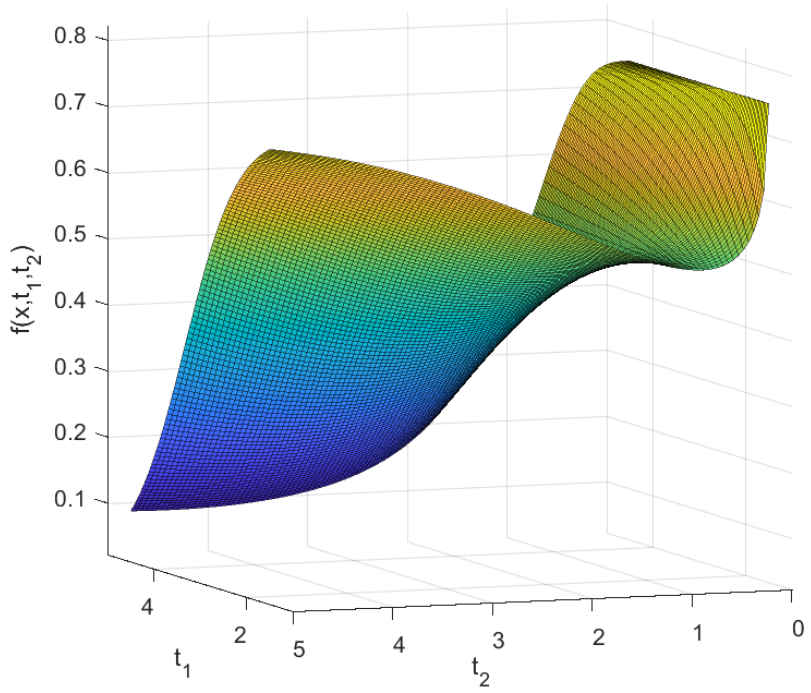

Figure 1: Example computer model output over the support of the calibration parameter t_c and the design parameter t_d.

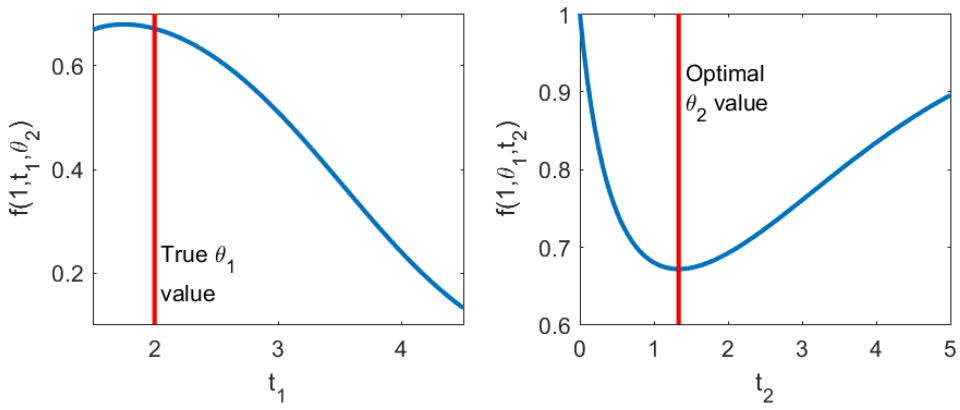

Figure 2: The lefthand plot shows the computer model output at $x = 1$ and optimal θ_d for each value of the calibration parameter t_c. The righthand plot show the model output at $x = 1, t_c = \theta_c$ for each value of the design parameter t_d.

4.1 Results

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

$$f_1(x, t_d) = \eta(x, \theta_c, t_d)\,(1 - a(x - .5)(x - 1)/x))$$
$$f_2(x, t_d) = \eta(x, \theta_c, t_d) - a(x - .5)(x - 1)\left(t_d - \frac{4}{3}\right)^2 + b$$
$$f_3(x, t_d) = \eta(x, \theta_c, t_d) + axt_d + b$$

Where a, b are constants which determine how severe the discrepancy is in each case. The function f_1 has a multiplicative discrepancy dependent only on x. This discrepancy does not affect the optimal value of t_d. The discrepancy of f_2 is additive, and is dependent upon both x and θ_c. Though this discrepancy can affect the optimal value of t_d, in the case that $\theta_c = 2$ (which is what we assume to be the truth) it does not. Thus under f_1 and f_2, it remains the case that the optimal value of t_d is $\theta_d = 4/3$. By contrast, f_3 has an additive discrepancy which does affect the optimal setting for t_d. For f_3, optimal t_d is dependent upon both the true value of θ_c and upon the value of a. For example, for $\theta_c = 2$ and $a = 0.055$, the optimal t_d is $\theta_d \approx 1$. Figure 3 shows the discrepancies for two different versions (corresponding to different settings of (a, b)) of each f_i.

We applied DCTO to each of seven cases: the non-discrepancy case, and the two different versions of each f_i shown in Figure 3. We found that in these cases, no appreciable

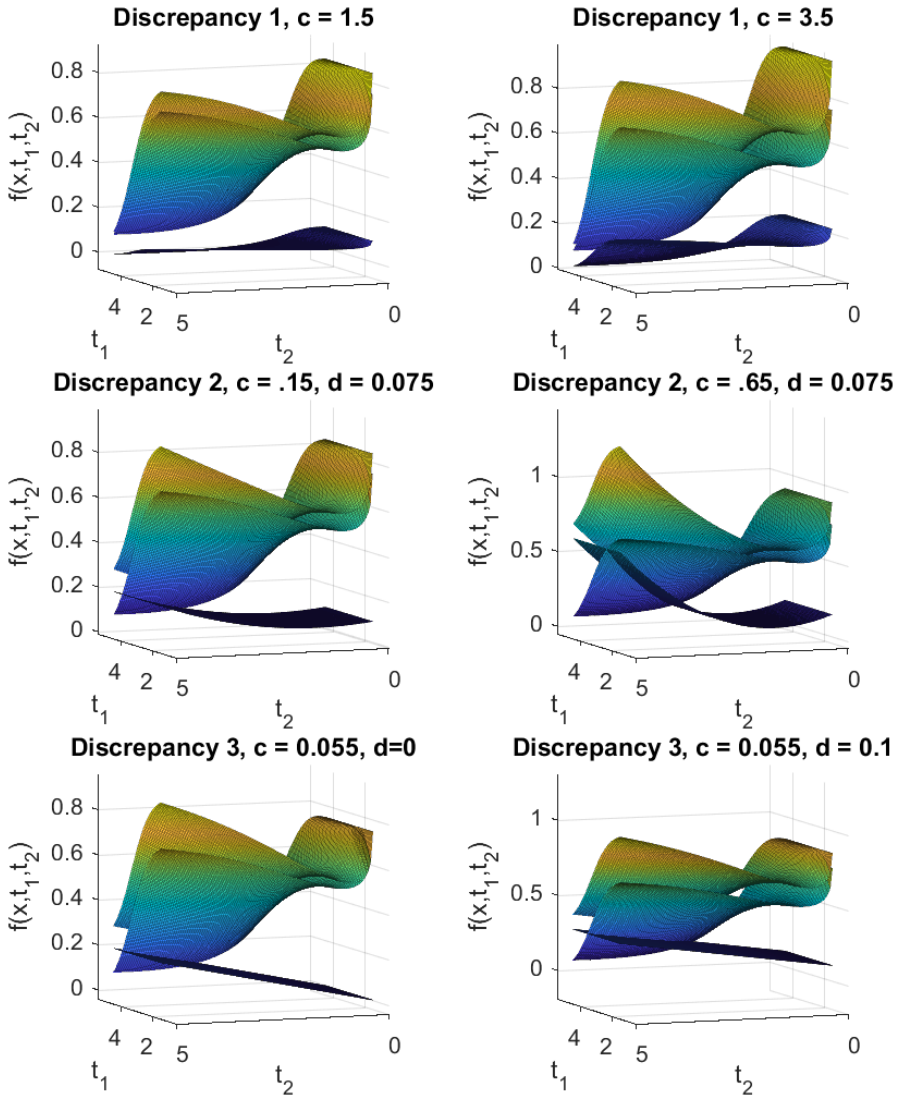

Figure 3: The i^{th} row shows f_i (the objective function with discrepancy), η (the computer model), and the discrepancy $f_i - \eta$, all at $x = 0.75$. In each row, a less aggressive version of the discrepancy appears on the left, and a more aggressive on the right. In each plot, the topmost surface is f_i, the middle surface is η, and the bottom surface is the discrepancy $f_i - \eta$.

difference resulted from the decision of whether or not to use an emulator (where the emulator was trained on a latin hypercube design of 250 points on the space of model inputs). Therefore, the results reported here do not employ an emulator. In each case, we gathered 50 "observations" of f_i on a latin hypercube design over the supports of x and θ_d, setting θ_c equal to its "true" value of 2. After standardizing the response to have mean 0 and standard deviation 1, we added i.i.d. N(0,0.05) noise to the response. We then carried out DCTO using Metropolis-Hastings-within-Gibbs MCMC, drawing 8000 samples each of $t_c, t_d, \boldsymbol{\rho}_\delta, \lambda_\delta, \sigma_d^2$, where $\boldsymbol{\phi}_\delta = (\boldsymbol{\rho}_\delta^T, \lambda_\delta)^T$. We modularized the analysis by drawing each of $\boldsymbol{\theta}_c, \boldsymbol{\rho}_\delta, \lambda_\delta$ using the likelihood based only on $\mathbf{y}_r$ rather than on $\mathbf{y}$.

In order to evaluate the success of the calibration component of DCTO, we also carried out a two-step procedure of using traditional KOH calibration of θ_1, followed by a second step using CTO to obtain a distribution of θ_2. The first step is essentially DCTO with $\mathbf{x_d}, \mathbf{y_d}$ as empty (null) vectors, and the second step uses the distribution obtained in the first step to estimate θ_2. Thus, the comparison between DCTO and KOH+CTO shows the difference between DCTO and performing CTO on a system which has been calibrated using traditional methods. Figure 4 shows the results for g_0, the case of no discrepancy. The two methods deliver comparable results, showing that combining KOH calibration and CTO design into DCTO does not undermine the performance of either task. Strong Bayesian learning has occurred for both parameters, in that the posterior distributions of θ_1, θ_2 are peaked around their true and optimal values, respectively. KOH gives a similar posterior for θ_1, showing that the expansion of DCTO to undertake design has not interfered with its calibration performance. The skewness apparent in the posterior distributions of

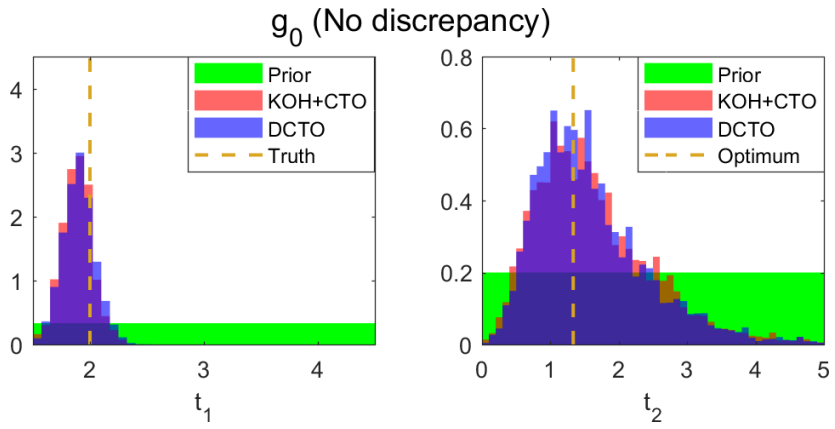

Figure 4: Prior and posterior distributions of the calibration parameter θ_1 and design parameter θ_2, along with their true/optimal values, for DCTO and KOH+CTO carried out when there is no discrepancy between the true system and the computer model.

θ_2 occur in all of the results gathered here, and is likely due to the shape of the objective function f, which increases sharply for $t_2 < \theta_2$ and increases much more gently for $t_2 > \theta_2$.

Figure 5 shows the results for g_1 at two settings of c, and Figure 6 shows the results for g_2 at two settings of (c, d). Somewhat counterintuitively, even stronger Bayesian learning occurs with respect to θ_1 in the case of g_1 than in the case of g_0, in each of the two settings of c. By contrast, and less surprisingly, the posterior distributions for θ_1 are somewhat wider in the case of g_2, for each of the two settings of (c, d). Nonetheless, the posterior distributions for g_2, as for g_0 and g_1, still peak at the true value of θ_1 and at the optimal value of θ_2 under DCTO. The case of g_2 with $(c, d) = (.65, .075)$ is the only case in which DCTO and KOH+CTO produce strikingly different results. Here, DCTO supplies a posterior distribution for θ_2 that peaks at the true optimum, while KOH+CTO produces a much wider distribution that fails to peak at the optimum. The reason for this difference is unclear. DCTO and KOH both have fairly high uncertainty in the posterior distribution of θ_1, which propogates to uncertainty in the posterior distribution of θ_2, since the optimal

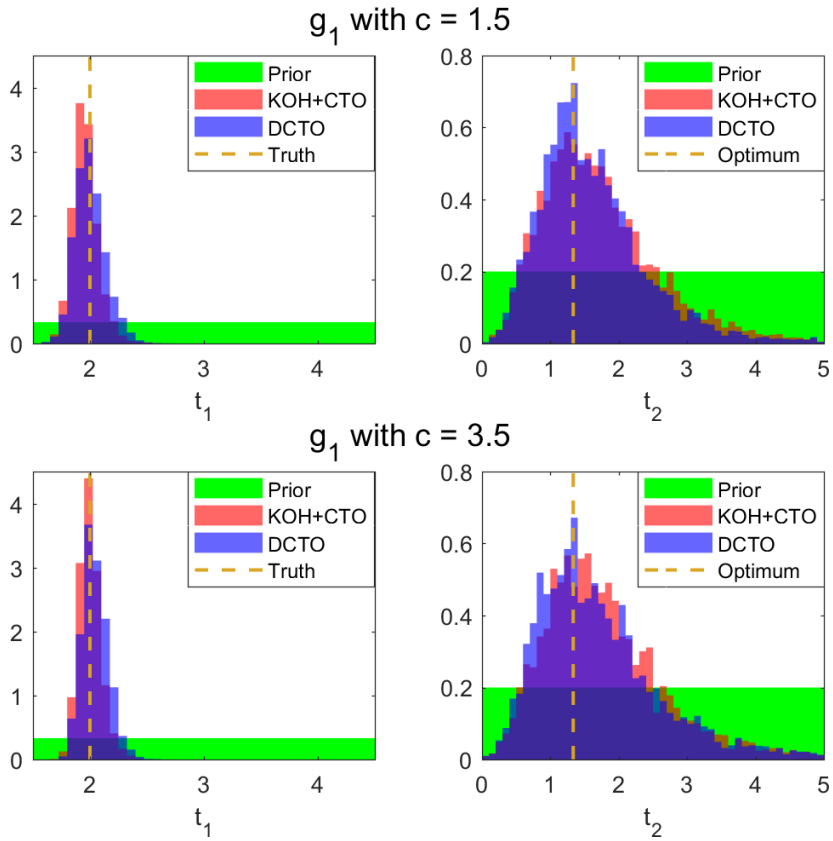

Figure 5: Prior and posterior distributions of the calibration parameter θ_1 and design parameter θ_2, along with their true/optimal values, for DCTO and KOH+CTO in the case of true systems g_1. The top row corresponds to a smaller discrepancy, with $c = 1.5$; the bottom row corresponds to a larger discrepancy, with $c = 3.5$.

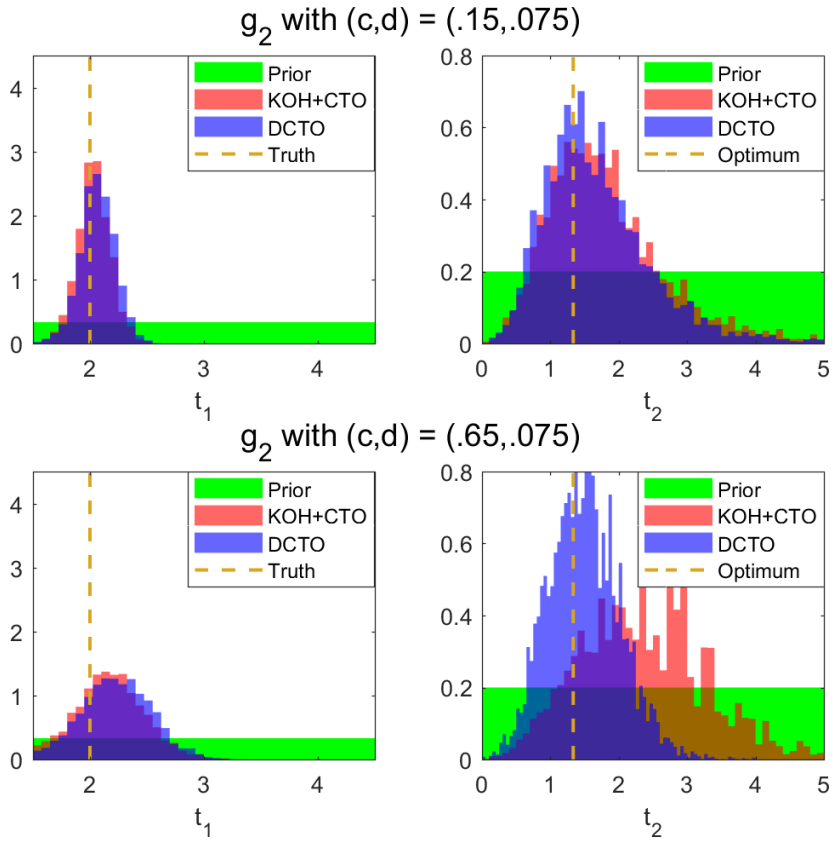

Figure 6: Prior and posterior distributions of the calibration parameter θ_1 and design parameter θ_2, along with their true/optimal values, for DCTO and KOH+CTO in the case of true systems g_2. The top row corresponds to a smaller discrepancy, with $(c, d) = (.15, .075)$; the bottom row corresponds to a larger discrepancy, with $(c, d) = (.65, .075)$.

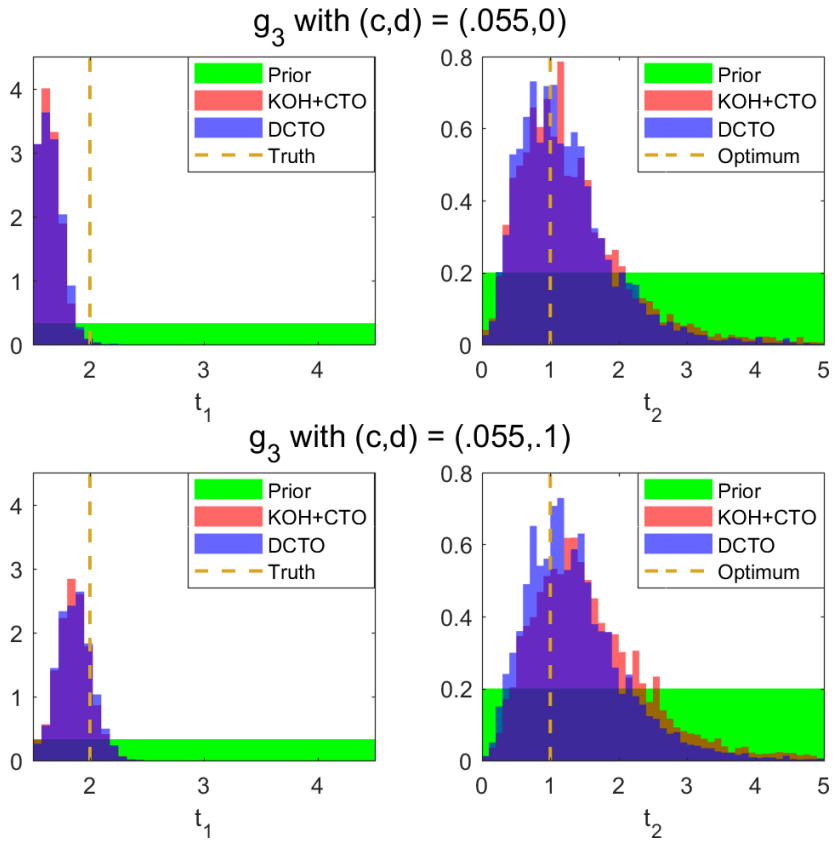

Figure 7: Prior and posterior distributions of the calibration parameter θ_1 and design parameter θ_2, along with their true/optimal values, for DCTO and KOH+CTO in the case of true systems g_3. The top row corresponds to a smaller discrepancy, with $(c, d) = (.055, 0)$; the bottom row corresponds to a larger discrepancy, with $(c, d) = (.055, .1)$.

θ_2 is dependent upon the true value of θ_1. It is possible that this affects KOH+CTO more than DCTO since DCTO is able to use the information contained in the true observations of g_2 to inform the design of θ_2, whereas under KOH+CTO the true observations are used only in calibration of θ_1 and not in the design of θ_2.

Matters change in the case of g_3, where the calibration procedure is somewhat unsuccessful for both DCTO and KOH+CTO. Figure 7 (upper left) shows that the true value of θ_1 is well into the tails of the posterior distributions. Surprisingly, increasing d from 0 to 0.1 and keeping $c = 0.055$, the results are significantly better, even though the discrepancy

in this case is larger. In the lower left of Figure 7, we see that the posterior distributions again peak sharply near the true value of θ_1. In all versions of the discrepancy function explored here, the posterior distribution under DCTO is roughly similar with respect to θ_2; wider than the posteriors for θ_1, but peaking near the optimal θ_2 even when this is not true of θ_1.

In order to achieve better results in the case of g_3 with $d = 0$, we attempted to place a more informative prior on the discrepancy function δ_1. We pursued two different strategies to do this. Firstly, we integrated the discrepancy $g_3 - f$ over the supports of x and t_2, finding the average value of the discrepancy. We then re-ran DCTO using $m_1(x, t_2) = \left(\int (g_3(z, \theta_1, w) - f(z, \theta_1, w))dzdw\right)/2.5$ as the (constant) mean of the GP prior on δ_1. This corresponds to a case in which one knows on average how far one's model tends to be from the true system. The results, however, were not appreciably different from the original results using a mean of 0 for the GP prior on δ_1. Secondly, we again ran DCTO using $m_1(x, t_2) = g_3(x, \theta_1, t_2) - f(x, \theta_1, t_2)$, so that the GP prior mean on δ_1 was the true discrepancy. This corresponds to a case in which one has (e.g. through extensive experimentation) a more thorough understanding of the form of the model's discrepancy with the true system. This second strategy proved fruitful, producing the results in Figure 8. KOH+CTO and DCTO were equally successful in calibration, but DCTO again outperforms KOH+CTO on design here, with a posterior which is both narrower and more centered on the true optimum. Given that the two design procedures perform similarly with an uninformative prior on δ_1, it's unclear why using an informative prior produces such different design results, especially since the informative prior has an almost identical

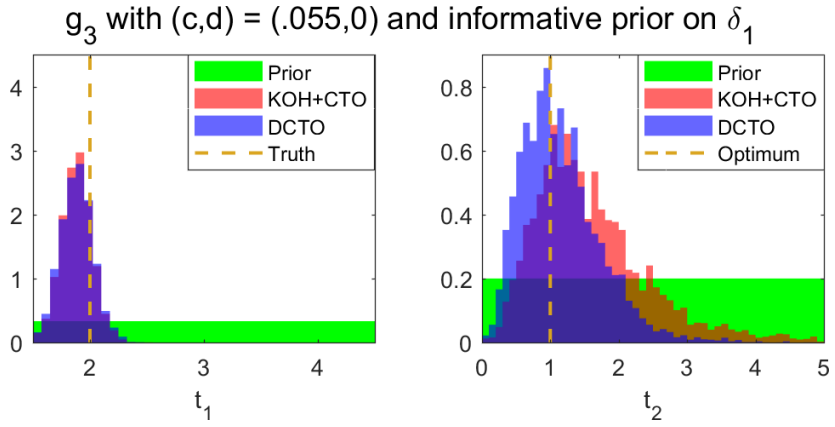

Figure 8: Prior and posterior distributions of the calibration parameter θ_1 and design parameter θ_2, along with their true/optimal values, for DCTO ond KOH in the case of true systems g_3 with $(c, d) = (.055, 0)$ and an informative prior for δ_1.

effect in improving the calibration results in the two cases.

The posterior predictive distributions under DCTO and KOH+CTO are very close under six of the eight discrepancies studied. Nonetheless, in most of those six cases DCTO enjoys very slightly narrower posterior predictive distributions. In the two cases under which DCTO and KOH+CTO diverge appreciably, DCTO enjoys much better results, with sharper distributions that center nearer to the true optimal model output. Figure 9 shows the posterior predictive distributions for g_0, the case of no model discrepancy, and for two settings of g_1, the case of multiplicative model discrepancy. Here, the posterior predictive distributions for DCTO and KOH+CTO overlap almost perfectly, except for the slightly higher peaks of DCTO.

Figure 10 shows the results for two settings of g_2, an additive discrepancy form. For the smaller discrepancy setting of g_2, the results are similar to those for g_0 and g_1, with DCTO enjoying slightly less uncertainty. The larger discrepancy setting for g_2 is the same case as that shown in Figure 6, where KOH+CTO performs well with respect to calibration

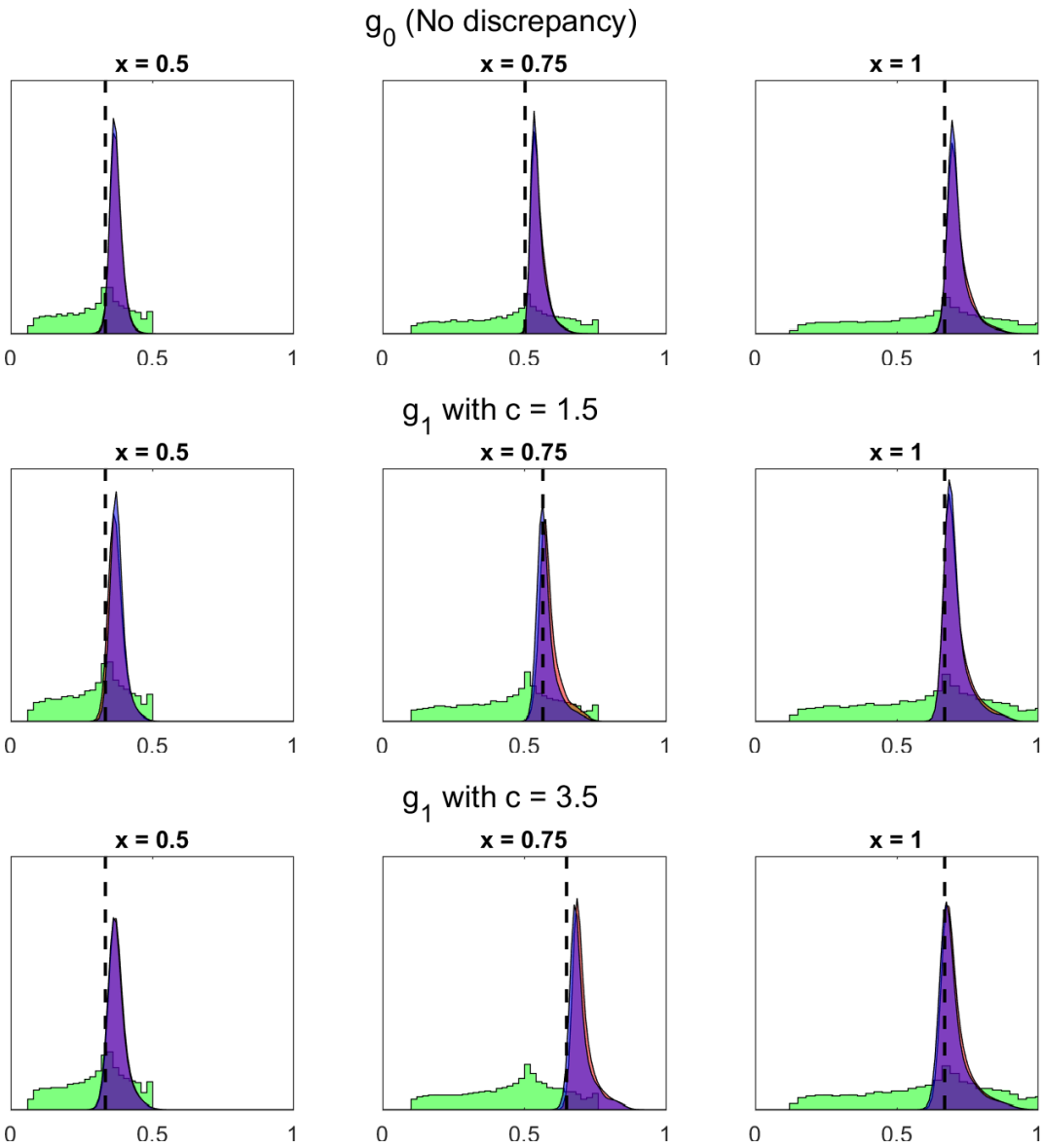

Figure 9: Prior (green) and posterior predictive distributions of the model output, along with the optimum, for DCTO and KOH+CTO at three values of the control input x. The red KOH+CTO and blue DCTO distributions overlap almost perfectly, but in all cases except the bottom middle plot, DCTO peaks just slightly above KOH+CTO.

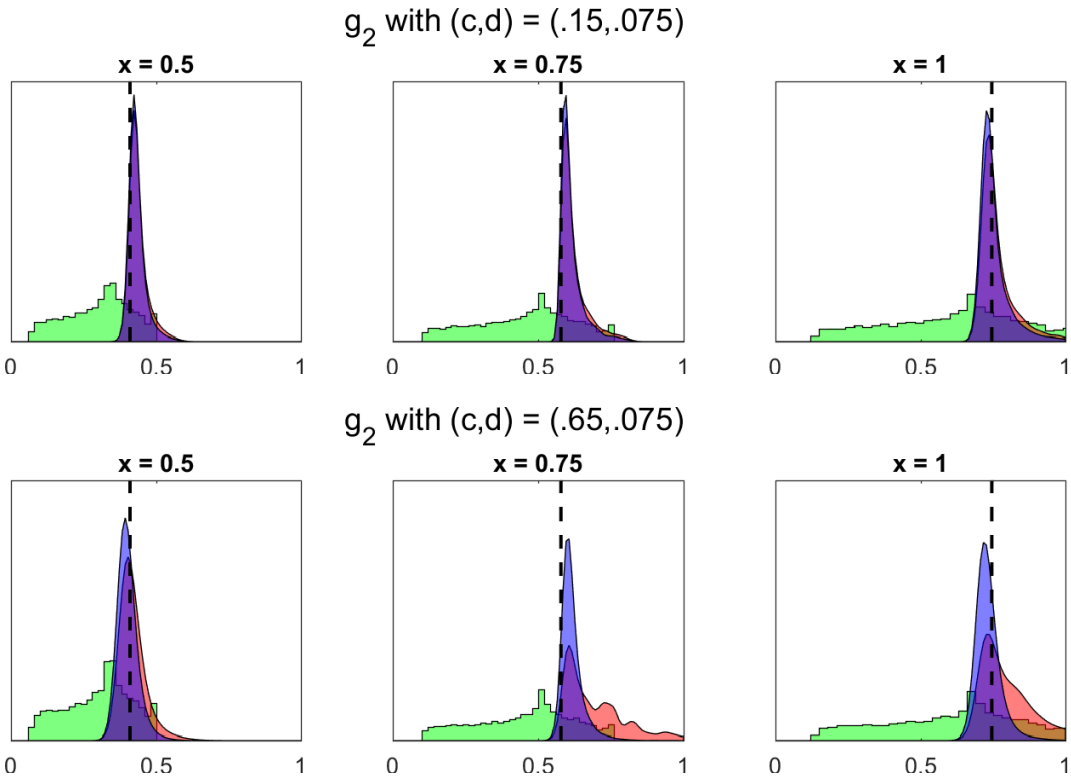

Figure 10: Prior (green) and posterior predictive distributions of the model output, along with the optimum, for DCTO and KOH+CTO at three values of the control input x.

but poorly with respect to design. Figure 10 shows that this poor design corresponds to wildly different predictive performance between DCTO and KOH+CTO, with KOH+CTO suffering from high uncertainty and strange multimodal behavior for some levels of the control input x. It is however notable that even here, the posterior predictive distribution's global mode still occurs at the same location as the global mode under DCTO, despite the fact that this is not true of the posterior distributions of the design parameter t_2 under the two techniques.

The additive discrepancy form of g_3 again has similar posterior predictive results under DCTO and KOH+CTO for both small and large versions of the discrepancy, except when an informative prior is supplied for the discrepancy function. Recall that the small discrepancy

version of g_3 is a case in which both DCTO and KOH performed poorly with respect to calibration, as shown in Figure 7. The top row of Figure 11 shows that this poor calibration does not result in similarly poor predictive performance. The improved calibration that results from applying an informative prior, as shown in Figure 8, inflates the posterior uncertainty surrounding the design variable t_2 for KOH+CTO, while slightly deflating the t_2 uncertainty for DCTO. Figure 11 (middle row) seems to show that this difference between KOH+CTO translates to similar differences in posterior predictive distributions when using the informative prior: slightly less uncertainty under DCTO, slightly more under KOH+CTO. However, despite the fact that both techniques experience dramatically better calibration under the informative prior than without it, there is not a similarly dramatic difference between the posterior predictive distributions with and without the informative prior, under either technique. The flexibility of the discrepancy term in CTO allows for accurate prediction even when the calibration is inaccurate.

The calibration, design, and predictive results for all seven cases (one case of zero discrepancy plus two versions each of three different forms of discrepancy) are summarized in Table 1. The top table shows the posterior variances of the calibration parameter t_1, the design parameter t_2, and the predictive distribution $\tilde{y}$ at control setting $x = 0.75$. The bottom table shows the absolute deviation of the resulting posterior means $\widehat{\theta_1}, \widehat{\theta_2}$, and $\hat{\tilde{y}}$ from their known true values. It appears that KOH produces narrower calibration posteriors than DCTO, as the variances for θ_1 are uniformly lower for KOH, if usually only slightly. However, when we instead look at $|\theta_1 - \widehat{\theta_1}|$, matters are less clear. DCTO has a lower calibration estimate absolute error in five of the eight cases studied here, and in

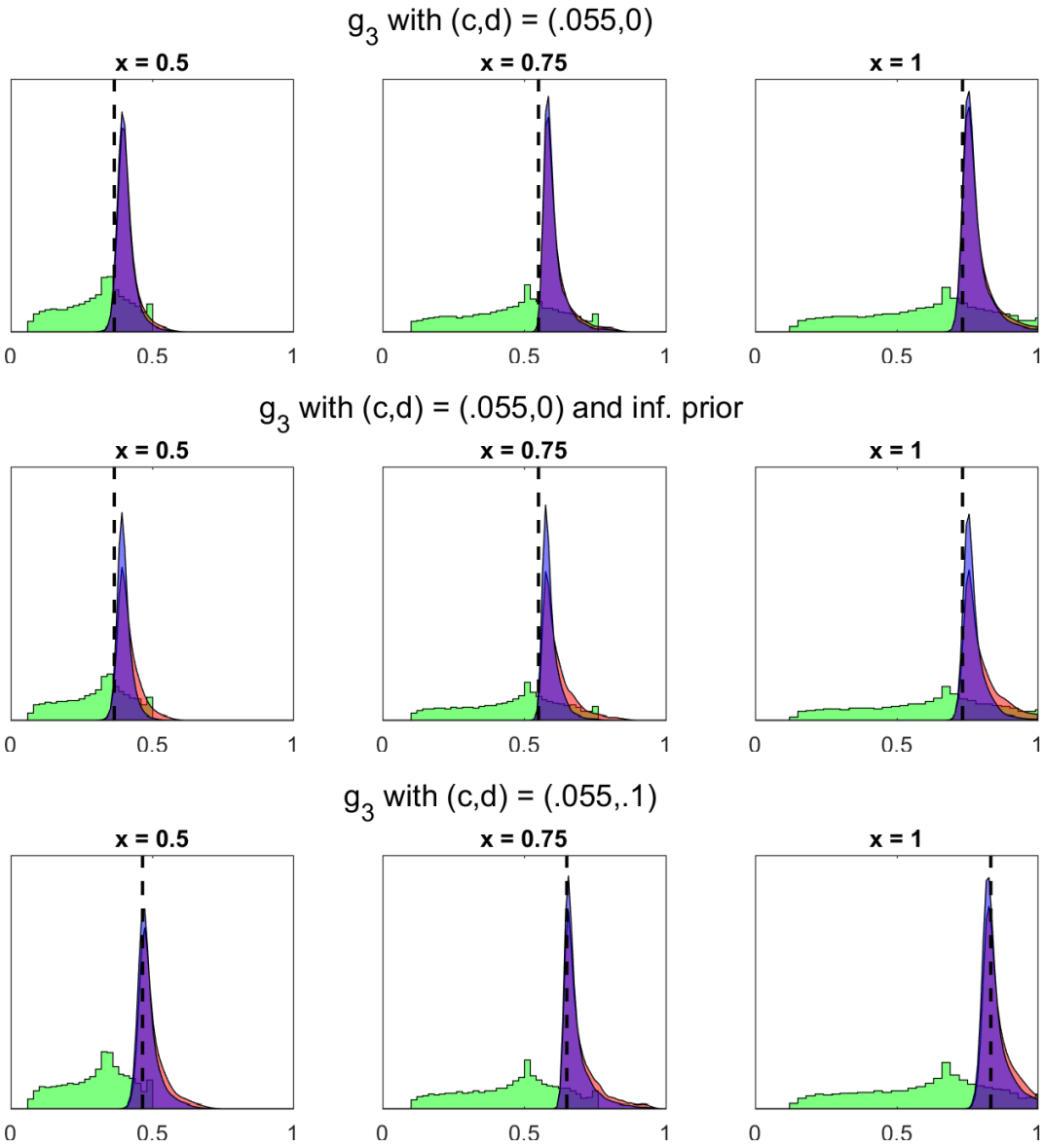

Figure 11: Prior (green) and posterior predictive distributions of the model output, along with the optimum, for DCTO and KOH+CTO at three values of the control input x.

	t_1 variance		t_2 variance	
Discrepancy	DCTO	KOH+CTO	DCTO	KOH+CTO
0	0.0198	0.0179	0.6636	0.7066
1, small	0.0181	0.0131	0.5881	0.7055
1, large	0.0129	0.0096	0.7676	0.7465
2, small	0.0233	0.0209	0.5849	0.7022
2, large	0.0896	0.0872	0.2711	0.7992
3, small	0.0108	0.0093	0.4903	0.6257
3, large	0.0216	0.0207	0.5228	0.7009
3, small, inf.	0.0201	0.0179	0.3330	0.6787

	$\lvert\theta_1 - \widehat{\theta_1}\rvert$		$\lvert\theta_2 - \widehat{\theta_2}\rvert$	
Discrepancy	DCTO	KOH+CTO	DCTO	KOH+CTO
0	0.0969	0.1124	0.2594	0.2838
1, small	0.0007	0.0427	0.2720	0.3816
1, large	0.0436	0.0001	0.3614	0.4361
2, small	0.0686	0.0353	0.3365	0.4664
2, large	0.2346	0.1808	0.1577	1.0270
3, small	0.3393	0.3464	0.2304	0.3090
3, large	0.1325	0.1403	0.3651	0.5781
3, small, inf.	0.1151	0.1157	0.1488	0.5921

Table 1: Posterior variances and mean absolute deviations for the calibration variable θ_1 and the design variable θ_2. The estimator $\widehat{\theta_i}$ is the posterior mean of t_i. The final line in each table gives the results for the use of an informative prior for the discrepancy of g_3 with $(c, d) = (0, .055)$.

general, the differences between the two procedures here is small. It thus appears that DCTO and KOH+CTO enjoy similar levels of success in calibration. Clearer differences emerge in looking at the performance of the two procedures with respect to the design variable. DCTO produces a smaller posterior variance in seven out of the eight cases, in two cases more than halving the variance produced under KOH+CTO. The difference is even more pronounced for $|\theta_2 - \widehat{\theta_2}|$ under the two procedures, as DCTO produces a lower absolute error in each of the eight cases, sometimes strikingly so. For example, in the large discrepancy version of g_2, DCTO produces an estimate with less than one sixth the absolute error of KOH+CTO. It appears that DCTO and KOH+CTO perform similarly with respect to calibration, but that DCTO produces superior results with respect to design.

References

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.