

# 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

March 8, 2019

## Abstract

Abstract

*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

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  $\theta$  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 The model

Let  $f$  be the model of interest, and partition its inputs as  $(x, \theta_1, \theta_2)$  where  $x$  denotes the known and/or controllable inputs,  $\theta_1$  denotes the parameters targeted for KOH calibration, and  $\theta_2$  denotes the input settings targeted for design via CTO. If  $f$  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  $f$  with mean  $m_0(x, t_1, t_2)$  and covariance function  $C_0((x, t_1, t_2), (x', t'_1, t'_2))$ . From here on in this discussion, assume that a GP surrogate is used for  $f$ . Typically, there will be some systematic discrepancy between the output of  $f$  and the true system, even if the true value of  $\theta_1$  is used. We model this discrepancy, between the true system and  $f$  at the true value of  $\theta_1$ , with another GP prior  $\delta_1$  having mean  $m_1(x, t_2)$  and covariance function  $C_1((x, t_2), (x', t'_2))$ . In addition to the discrepancy between  $f$  and the true system, there may also be a systematic discrepancy between the true system and our performance/cost targets, particularly if the latter are especially optimistic. We model this discrepancy using a GP prior with mean  $m_2(x)$  and covariance function  $C_2(x, x')$ . Notice that this discrepancy is between the true system at optimal settings for  $\theta_2$  and  $f$  using the true value of  $\theta_1$  and the optimal settings for  $\theta_2$ . In addition to systematic discrepancy between  $f$  and reality, measurement error must be included in the model. Thus let  $\epsilon \sim N(0, \sigma^2)$  denote the (known) i.i.d. variance of (real) observations at any given values of  $x, t_2$ . Finally, we assume that  $\eta, \delta_1, \delta_2$ , and  $\epsilon$  are all mutually independent.

A collection of simulation runs is needed to train the GP code surrogate. Let  $(\mathbf{x}_s, \mathbf{t}_{1s}, \mathbf{t}_{2s})$  be the design matrix for these simulation runs, and let  $\mathbf{y}_s$  denote the output of these runs. Similarly, let  $\mathbf{y}_r$  be the real observations at  $\mathbf{x}_r, \theta_{2r}$ , and let  $\mathbf{y}_d$  be a set of artificial “observations” reflecting our cost and performance targets at control settings  $\mathbf{x}_d$ . Call  $y_d$  the *target outcomes*. Finally, let  $\mathbf{y} = (\mathbf{y}_s, \mathbf{y}_r, \mathbf{y}_d)^T$ . Then it follows that  $\mathbf{y} \sim N(\mathbf{m}, \mathbf{C})$ ,

where

$$\mathbf{m} = \begin{pmatrix} m_0(\mathbf{x}_s, \mathbf{t}_{1s}, \mathbf{t}_{2s}) \\ m_0(\mathbf{x}_r, \mathbf{1}\theta_1, \mathbf{t}_{2r}) + m_1(\mathbf{x}_1, \mathbf{t}_{2r}) \\ m_0(\mathbf{x}_d, \mathbf{1}\theta_1, \mathbf{1}\theta_2) + m_1(\mathbf{x}_d, \mathbf{1}\theta_2) + m_2(\mathbf{x}_d) \end{pmatrix},$$

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

$$\mathbf{C}_{11} = C_0((\mathbf{x}_s, \mathbf{t}_{1s}, \mathbf{t}_{2s}), (\mathbf{x}_s, \mathbf{t}_{1s}, \mathbf{t}_{2s}))$$

$$\mathbf{C}_{21} = C_0((\mathbf{x}_s, \mathbf{t}_{1s}, \mathbf{t}_{2s}), (\mathbf{x}_r, \mathbf{1}\theta_1, \mathbf{t}_{2r}))$$

$$\mathbf{C}_{31} = C_0((\mathbf{x}_s, \mathbf{t}_{1s}, \mathbf{t}_{2s}), (\mathbf{x}_d, \mathbf{1}\theta_1, \mathbf{1}\theta_2))$$

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

$$\mathbf{C}_{22} = C_0((\mathbf{x}_r, \mathbf{1}\theta_1, \mathbf{t}_{2r}), (\mathbf{x}_r, \mathbf{1}\theta_1, \mathbf{t}_{2r})) + C_1((\mathbf{x}_r, \mathbf{t}_{2r}), (\mathbf{x}_r, \mathbf{t}_{2r})) + \sigma_2 \mathbf{I}$$

$$\mathbf{C}_{32} = C_0((\mathbf{x}_r, \mathbf{1}\theta_1, \mathbf{t}_{2r}), (\mathbf{x}_d, \mathbf{1}\theta_1, \mathbf{1}\theta_2)) + C_1((\mathbf{x}_r, \mathbf{t}_{2r}), (\mathbf{x}_d, \mathbf{1}\theta_2))$$

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

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

$$\mathbf{C}_{33} = C_0((\mathbf{x}_s, \mathbf{1}\theta_1, \mathbf{1}\theta_2), (\mathbf{x}_s, \mathbf{1}\theta_1, \mathbf{1}\theta_2)) + C_1((\mathbf{x}_s, \mathbf{1}\theta_2), (\mathbf{x}_s, \mathbf{1}\theta_2)) + C_2(\mathbf{x}_d, \mathbf{x}_d)$$

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, DCTO generalizes the KOH framework.

### 3 Example with simulated data

Consider the function of three inputs  $f(x, t_1, t_2) = x/(t_2^{t_1-1} \exp(-0.75t_2) + 1)$ . Figure ?? shows the output of this function for  $x = 1$  over the range  $(t_1, t_2) \in [1.5, 4.5] \times [0, 5]$ . We arbitrarily set  $\theta_1 = 2$  to be the “true” value of  $t_1$ . For any value of  $x$  and  $t_1$ , the optimal (minimizing) value of  $t_2$  is  $(4/3)(t_1 - 1)$ , so we have  $\theta_2 = 4/3$ . Figure ?? shows the locations of the true and optimal values (respectively) of  $\theta_1$  and  $\theta_2$ . There it is clear that the true value of  $\theta_1$  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  $f$  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.

#### 3.1 Results

We used DCTO on four versions of the problem. First, we assumed that  $f$  is free from discrepancy – i.e. that  $f(x, \theta_1, t_2)$  is an unbiased estimator of the “true” system  $g(x, t_2)$ . The other three versions each assume that  $f$  suffers from some form of discrepancy. Let  $g_1, g_2, g_3$  denote the “true” systems in these three cases. We set  $g_1(x, t_2) = f(x, \theta_1, t_2)$ . To train the GP code surrogate, we drew a latin hypercube design of 250 points in the space of the model inputs.

## 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.