Computer model calibration for design, with an application to wind turbine blades

Carl Ehrett[1,2], Andrew Brown[1,2], Sez Atamturktur[1,3], Christopher Kitchens[1,4], Mingzhe Jiang[1,4], Caleb Arp[1,4], Evan Chodora[1,3]

[1]Clemson University, [2]Department of Mathematical Sciences, [3]Glenn Department of Civil Engineering, [4]Chemical and Biomolecular Engineering

Computer model calibration

- Computer models may include unknown inputs that must be estimated[1]. These inputs are often estimated by combining simulator output with field data.
- Calibration is ordinarily thought of as bringing a computer model into agreement with reality.

Full model

- Where f is the true system, η the computer model of f, δ the discrepancy $f-\eta$, $\boldsymbol{\theta}$ the true values of the calibration parameters, $\mathbf{x}$ the control inputs, and $\boldsymbol{\epsilon}$ error, we have $f(\mathbf{x},\boldsymbol{\theta}) \equiv f(\mathbf{x}) = \eta(\mathbf{x},\boldsymbol{\theta}) + \delta(\mathbf{x}) + \boldsymbol{\epsilon}$.
- When η is computationally expensive, we use a mean-zero Gaussian process (GP) emulator as a code surrogate. δ is also modeled using a mean-zero GP. Both GPs use a power product exponential covariance function: respectively (where $\mathbf{x} \in \mathbb{R}^p$, $\boldsymbol{\theta} \in \mathbb{R}^q$),

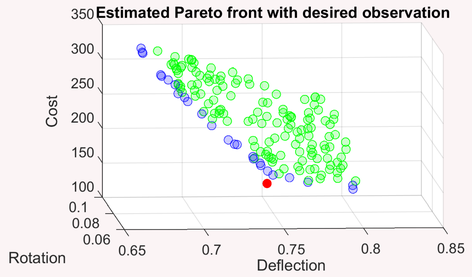
$$C_\eta((\mathbf{x},\boldsymbol{\theta}),(\mathbf{x}',\boldsymbol{\theta}')) = \frac{1}{\lambda_\eta}\prod_{k=1}^{p}\left(\rho_k^\eta\right)^{(x_k-x_k')^2} \times \prod_{j=1}^{q}\left(\rho_{p+j}^\eta\right)^{\left(\theta_j-\theta_j'\right)^2},$$

$$C_\delta(\mathbf{x},\mathbf{x}') = \frac{1}{\lambda_\delta}\prod_{k=1}^{p}\left(\rho_k^\delta\right)^{(x_k-x_k')^2}$$

  We use the MLEs of $\lambda_\eta, \boldsymbol{\rho}^\eta$, set $\rho_i^\delta \sim \text{Beta}(1, 0.3)$ for all i, and set an informative Gamma prior on λ_δ based on what is known *a priori* about the system optimum.
- Let $\boldsymbol{\eta}$ be a vector of univariate observations of the computer model at points $\{(\mathbf{x}_i,\boldsymbol{\theta}_i)\}_{i=1}^m$ and $\mathbf{y}$ be a set of "desired observations" at points $\{(\mathbf{x}_i,\boldsymbol{\theta}_i)\}_{i=m+1}^{m+n}$. Where $\mathcal{D} = [\boldsymbol{\eta}^T, \mathbf{y}^T]^T$, we have $\mathcal{D}|\boldsymbol{\theta},\lambda_\delta,\boldsymbol{\rho}^\delta \sim \text{N}(0, \mathbf{C}_\mathcal{D})$, and $\pi(\boldsymbol{\theta},\lambda_\delta,\boldsymbol{\rho}^\delta|\mathcal{D}) \propto \pi(\mathcal{D}|\boldsymbol{\theta},\lambda_\delta,\boldsymbol{\rho}^\delta)\times\pi(\boldsymbol{\theta})\times\pi(\lambda_\delta)\times\pi(\boldsymbol{\rho}^\delta)$. where $\mathbf{C}_\mathcal{D}$ is given by C_η, plus C_δ in the submatrix corresponding to $\mathbf{y}$. We explore this via MCMC.

Choosing target observations

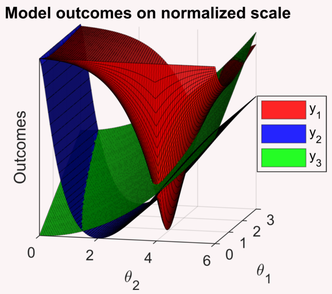
- To optimize, we need targets exceeding achievable results. But for identifiability, targets should be near the model range. To satisfy these constraints, we need a rough estimate of the Pareto front.
- We achieve this by a preliminary round of calibration, with a weak prior on λ_δ. This exploits problems of identifiability in the Kennedy-O'Hagan calibration framework[1] to explore the Pareto front.
- Using the results of preliminary calibration, we select a performance target near the model range.



Central idea

Researchers look to computer experiments where physical experimentation is difficult or impossible[2, 3]. Previous explorations of computer model calibration have approached calibration as a matter of bringing a computer model into agreement with physical reality[4, 1, 5, 6]. **In the present work, we consider computer model calibration as a method for design.** Under this framework, we calibrate a computer model not using physical experimental data, but rather using "desired observations" which describe the performance one hopes to achieve in the simulated system.

Artificial system example



- Consider $f : [1.95, 2.05] \times [0, 3] \times [0, 6] \to \mathbb{R}^3$ where the latter two inputs are treated as calibration parameters. We seek to minimize the three model outputs:

$$f_1(x,\theta_1,\theta_2) = \left(\theta_1 \exp\left(-\left(\theta_1 + |\theta_2 - \frac{\pi x}{2}|\right)\right) + 1\right)^{-1}$$

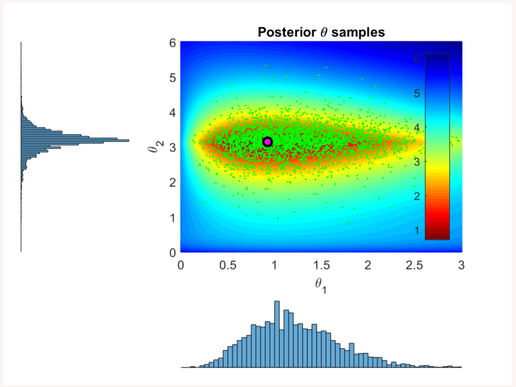
$$f_2(x,\theta_1,\theta_2) = \left(\theta_2^{x-1}\exp\left(-0.75\theta_2\right) + 1\right)^{-1}$$

$$f_3(x,\theta_1,\theta_2) = 15 + 2\theta_1 + \frac{\theta_2^2}{4}$$

- Initially, the performance target of $[0, 0, 0]$ was chosen. To improve the identifiability of the optimal region, we used a preliminary round of calibration to update this to a target of $[0.71, 0.71, 17.92]$, constant w.r.t. x.

Gaussian process emulator

Gaussian processes (GPs) can be thought of as random functions which are generalizations of multivariate normal random variables[7]. A GP is fully characterized by its mean function $\mu : D \to R$ and covariance function $V : D \times D \to R$, where D is the domain of the process.

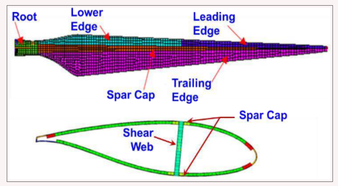


GPs are advantageous for emulating computationally expensive deterministic computer code[2, 3] because

- Use of a GP does not require detailed foreknowledge of the approximate parametric form of the model; we

Finite element smulator

We rely on a finite element simulator of the blade cost and performance.



- Calibration inputs: volume fraction, thickness of blade material. Control input: temperature.
- Outputs are tip deflection, rotation, and cost; the design goal is to minimize these.
- Model utilizes `ANSYS` simulation software; computation cost is too high for use in MCMC.

Emulator implementation

Model inputs are a dummy input x_1 (to convert trivariate output into univariate), temperature x_2, volume fraction x_3 and thickness x_4. For our GP emulator prior we use the covariance function

$$V(\mathbf{x},\mathbf{x}') = \frac{1}{\lambda}\exp\{-\sum_{i=1}^{4}\beta_i(x_i - x_i')^2\}$$

where $\lambda, \boldsymbol{\beta}$ were estimated using gradient methods to find their MLEs. A prior mean $\mu(\mathbf{x}) = 0$ was also used. Observations $\eta(\mathbf{x}_i)$, $i = 1, \cdots, 504$ were collected from the simulation model using a LHC design. The updated (posterior) GP emulator has mean given by

$$\mu^*(\mathbf{x}) = v(\mathbf{x})^T\Sigma^{-1}\boldsymbol{\eta}$$

where $v(\mathbf{x}) = (V(\mathbf{x},\mathbf{x}_1),\cdots,V(\mathbf{x},\mathbf{x}_{504}))^T$, Σ is a matrix with $\Sigma_{i,j} = V(\mathbf{x}_i,\mathbf{x}_j)$, and $\boldsymbol{\eta} = (\eta(\mathbf{x}_1),\cdots,\eta(\mathbf{x}_{504}))^T$. The updated covariance:

$$V^*(\mathbf{x},\mathbf{x}') = V(\mathbf{x},\mathbf{x}') - v(\mathbf{x})^T\Sigma^{-1}v(\mathbf{x}')$$

Desired data

We calibrate the model to "desired data" which reflect extremely low tip deflection, rotation, and cost. Because we are antecedently ignorant of how close the model can come to our desired data, we place an improper $1/\sigma^2$ prior on the observation variance of each model output. We considered a range of desired outcomes.

MCMC implementation

- Calibrate volume fraction and thickness to the desired data. Set a uniform prior over each.
- Each iteration of the MCMC, draw new values for x_3, x_4, and the observation variances.
- Where $\mathbf{y}$ is the desired data, $\mathcal{D} = (\mathbf{y},\boldsymbol{\eta})^T$ and $\Sigma_\mathcal{D} = \text{Var}(\mathcal{D})$, the likelihood of $\mathcal{D}$ is

$$L(\mathcal{D}|x_3,x_4,\boldsymbol{\lambda},\boldsymbol{\beta},\Sigma_\mathcal{D}) = |\Sigma_\mathcal{D}|^{-\frac{1}{2}}\exp\{-\frac{1}{2}\mathcal{D}^T\Sigma_\mathcal{D}^{-1}\mathcal{D}\}$$

  hence[6] the full posterior density is:

$$\pi(x_3,x_4,\sigma_d^2,\sigma_r^2,\sigma_c^2) \propto \frac{L(\mathcal{D}|x_3,x_4,\boldsymbol{\lambda},\boldsymbol{\beta},\Sigma_\mathcal{D})}{\sigma_d^2\sigma_r^2\sigma_c^2}$$

- Eliminate boundary constraints: $z_i = \text{logit}(x_i)$, $\tau_j = \log(\sigma_j^2)$ for $i = 3, 4$, $j = d, r, c$.
- We use the Metropolis-Hastings algorithm[8]; we set normal proposals on $z_i^{(n)}|z_i^{(n-1)}$ and $\tau_j^{(n)}|\tau_j^{(n-1)}$, $\forall i \forall j$.
- In burn-in, the proposal distribution adapts using the sample covariance of previous draws, achieving optimal acceptance ratios ($\sim 40\%$).

Results

- Posterior means are sensitive to choice of desired data. Therefore it is advisable to consider a surface over a range of desired data.
- Where desired output is insufficiently ambitious, the MCMC output reflects this as a lack of identifiability in the calibration inputs.

References

[1] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *JRSS: Series B (Statistical Methodology)*, vol. 63, pp. 425–464, aug 2001.

[2] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–423, 1989.

[3] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. New York: Springer, 2003.

[4] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu, "A Framework for Validation of Computer Models," *Technometrics*, vol. 49, pp. 138–154, may 2007.

[5] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal on Scientific Computing*, vol. 26, pp. 448–466, jan 2004.

[6] B. Williams, D. Higdon, J. Gattiker, L. Moore, M. McKay, and S. Keller-McNulty, "Combining experimental data and computer simulations, with an application to flyer plate experiments," *Bayesian Analysis*, vol. 1, pp. 765–792, dec 2006.

[7] A. O'Hagan and J. F. C. Kingman, "Curve Fitting and Optimal Design for Prediction," 1978.

[8] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, apr 1970.

Contact Information

Carl Ehrett
Email: cehrett@clemson.edu