# Simultaneous Determination of Tuning and Calibration Parameters for Computer Experiments

**Gang Han**[1], **Thomas Santner**[1], **and Jeremy Rawlinson**[2]

[1] Department of Statistics, The Ohio State University

[2] College of Veterinary Medicine, Cornell University

**Joint Statistical Meetings**

**Denver, Colorado**

**August 06, 2008**

# Outline

**I. Introduction and a motivating example**

**II. A Bayesian/distance method for simultaneous tuning and calibration**

**III. Conclusion**

# I. Introduction

• Computer experiments are used as surrogates for many physical experiments.

• Computer codes used for computer experiments have running times that can range from minutes to days .

• Some types of inputs to computer experiments
– Control variables: design inputs for both the computer and the physical experiments; e.g., engineering design inputs.
– Tuning parameters and calibration parameters.

- Tuning parameters

  1. Tuning parameters are the numerical values used by the algorithms implementing a mathematical model.
  2. Tuning parameters have *no* physical meaning in an associated physical experiment.
  3. The goal of tuning a computer code is to select values of the tuning parameter(s) to "best" match the physical experiment outcomes.
  4. Example: certain coefficients in differential equations.

- Calibration parameters

  1. Calibration parameters are controllable inputs to a computer code and are unknown or unmeasured during the running of an associated physical experiment.

  2. Calibration parameters typically have a meaning in the physical experiment.

  3. The goal of calibration is to make inference about the true calibration parameter value(s).

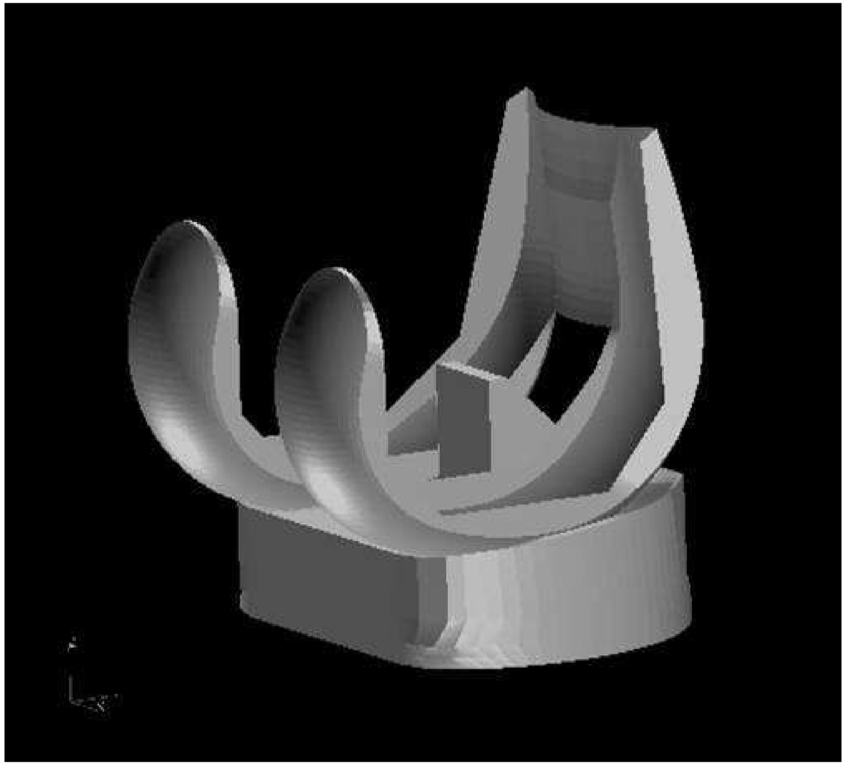
  4. Example: friction at the bone-prosthesis interface in a biomechanics application.

# Goal

To propose a method for simultaneous setting calibration and tuning parameters.

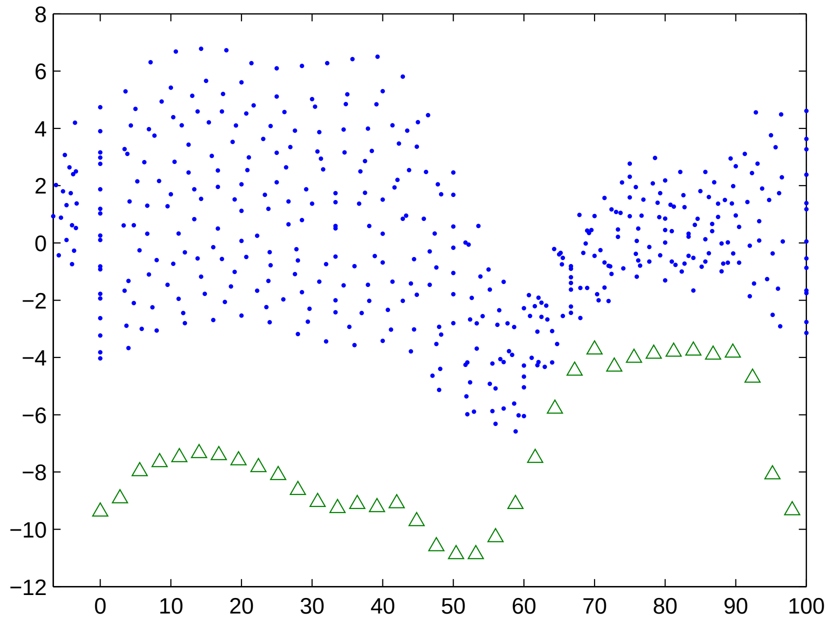
# A Motivating Example

• A finite-element analysis computer code simulating wear/movement of knees in a mechanical testing device.



- Response: anterior-posterior displacement **(APD)** of tibial tray relative to femoral component.
- Control variable: percentile of the **gait** cycle.
- Tuning parameter: **discretization** of loading curve for gait.
- Calibration parameter: **initial position** of femoral component with respect to tibial tray.

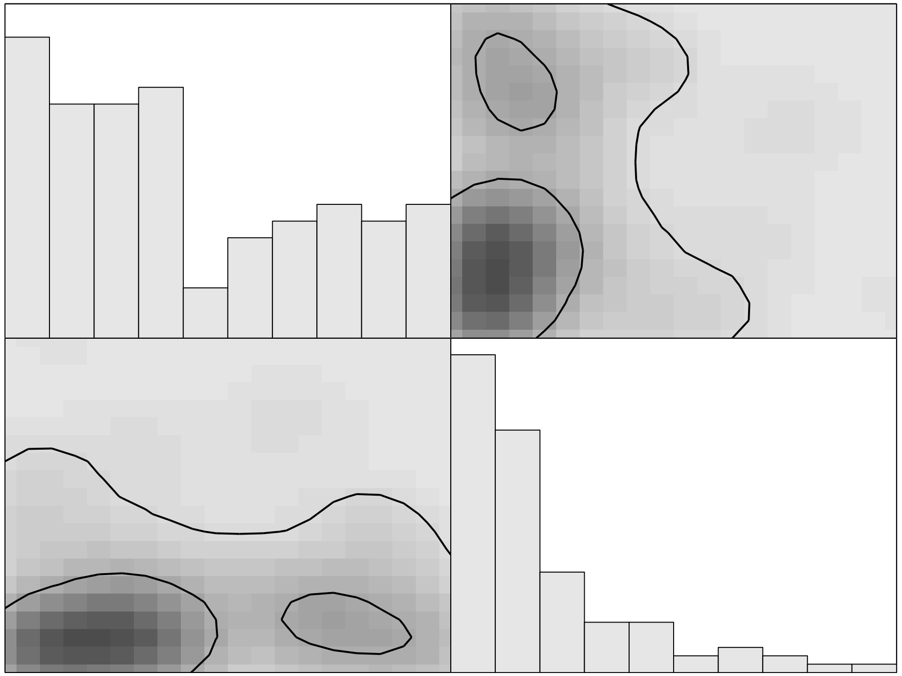
- Computer experiment: 438 runs; physical experiment: 36 runs.




**APD** from the knee simulator (triangles) and the FEA computer code (dots) vs **Gait cycle percentile** .

• One approach: treat both load discretization and initial position as calibration parameter; e.g., using the Bayesian approach of Higdon et. al. (2004).

Example: using Truncated Normal$(0.5, 10^2)$ (TN$(0.5, 10^2)$) priors on load discretization and initial position gives posteriors.



Marginal and joint simulated posterior distributions of **load discretization** (upper left) and **initial position** (lower right).

## II. A Bayesian/distance method for simultaneous tuning and calibration

• Our simultaneous calibration and tuning Bayesian model builds on Kennedy and O'Hagan, 2001, (*JRSSB*), Park, 1991, (*Ph.D. Thesis*), Higdon, et. al., 2004, (*SIAM Journal of Scientific Computing*), and Higdon, et. al., 2008, (*JASA*).

# Notation

- $\boldsymbol{x}$ == control variable; $\boldsymbol{t}$ == tuning parameter; $\boldsymbol{c}$ == calibration parameter; $\boldsymbol{\theta}_c$ == the true value of $\boldsymbol{c}$.
- $\{(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i), y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)\}_i^{n_s}$ == training data from the computer experiment.
- $\{(\boldsymbol{x}_j^p), y^p(\boldsymbol{x}_j^p)\}_j^{n_p}$ == training data from physical experiment
- $(n_s, n_p)$ == number of runs of computer and physical experiments.
- $\boldsymbol{y}^s = \left(y^s(\boldsymbol{x}_1^s, \boldsymbol{c}_1, \boldsymbol{t}_1), \ldots, y^s(\boldsymbol{x}_{n_s}^s, \boldsymbol{c}_{n_s}, \boldsymbol{t}_{n_s})\right)^\top$.
- $\boldsymbol{y}^p = \left(y^p(\boldsymbol{x}_1^p), \ldots, y^p(\boldsymbol{x}_{n_p}^p)\right)^\top$.

- **The idea**

1. Model the computer experiment $y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ as a draw from a Gaussian process $Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ and the discrepancy $\delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = E_\epsilon(Y^p(\boldsymbol{x})) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ as a draw from a Gaussian process $\Delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$.

- **The idea**

1. Model the computer experiment $y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ as a draw from a Gaussian process $Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ and the discrepancy $\delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = E_\epsilon(Y^p(\boldsymbol{x})) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ as a draw from a Gaussian process $\Delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$.

2. $$\boxed{Y^p(\boldsymbol{x}) = Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}^\star) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}^\star) + \epsilon(\boldsymbol{x})}$$

3. The three processes $Y^s(\cdot)$, $\Delta(\cdot)$, and the noise are **independent** .

# Modeling Computer Experiment Output

- $Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = \boldsymbol{\beta}_Z^\top \boldsymbol{f}_Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) + Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$.
- $\boldsymbol{\beta}_Z^\top \boldsymbol{f}_Z(\boldsymbol{x},\boldsymbol{c}, \boldsymbol{t})$: $\boldsymbol{f}_Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is the *known* regression coefficient vector and $\boldsymbol{\beta}_Z$ is the *unknown* regression parameter vector.
- $Z(\cdot)$ is a stationary Gaussian stochastic process with mean 0 and variance $\sigma_Z^2$. The correlation between $Z(\boldsymbol{x}_1, \boldsymbol{c}_1, \boldsymbol{t}_1)$ and $Z(\boldsymbol{x}_2, \boldsymbol{c}_2, \boldsymbol{t}_2)$.

$$\begin{aligned} & R_Z\left((\boldsymbol{x}_1, \boldsymbol{c}_1, \boldsymbol{t}_1), (\boldsymbol{x}_2, \boldsymbol{c}_2, \boldsymbol{t}_2)\right) \\ = \; & \prod_i \rho_{Z,x,i}^{4\times(x_{1,i}-x_{2,i})^2} \times \prod_j \rho_{Z,c,j}^{4\times(c_{1,j}-c_{2,j})^2} \times \prod_k \rho_{Z,t,k}^{4\times\left(t_{1,k}-t_{2,k}\right)^2}, \end{aligned}$$

where all $\rho \in [0, 1]$.

## Modeling the bias and the random noise processes

- $\Delta(\cdot)$ is a Gaussian stochastic process with mean $\boldsymbol{\beta}_\Delta^\top \boldsymbol{f}_\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ and variance $\sigma_\Delta^2$.

- The correlation between $\Delta(\boldsymbol{x}_1, \boldsymbol{\theta}_c, \boldsymbol{t}_1)$ and $\Delta(\boldsymbol{x}_2, \boldsymbol{\theta}_c, \boldsymbol{t}_2)$ is

$$\begin{aligned} & R_\Delta\left((\boldsymbol{x}_1, \boldsymbol{\theta}_c, \boldsymbol{t}_1), (\boldsymbol{x}_2, \boldsymbol{\theta}_c, \boldsymbol{t}_2)\right) \\ = \; & \prod_i \rho_{\Delta,x,i}^{4\times(x_{1,i}-x_{2,i})^2} \times \prod_k \rho_{\Delta,t,k}^{4\times(t_{1,i}-t_{2,i})^2}, \end{aligned}$$

  where $\{\rho_{\Delta,x,i}\}_i$ and $\{\rho_{\Delta,t,k}\}_k$ are correlation parameters.

- Model measurement error $\epsilon(\boldsymbol{x})$ as independent white noise process $(0, \sigma_\epsilon^2)$.

## Priors and Implementations

| **Model parameter** | **Prior distribution** | **Support** |
|---|---|---|
| All elements of $\boldsymbol{\theta}_c$ | $\mathrm{TN}(0.5, 2^2)$ | $[0, 1]$ |
| All elements of $\boldsymbol{\rho}_z$ and $\boldsymbol{\rho}_\Delta$ | $\mathrm{Beta}(1, 0.5)$ | $(0, 1)$ |
| $\sigma_z^2$ | $\mathrm{IG}(10, 0.1/\widehat{\sigma_s^2})$ | $(0, +\infty)$ |
| $\sigma_D^2$, if $\widehat{\sigma_s^2} > \widehat{\sigma_p^2}$ | $\mathrm{IG}(1, 100/\widehat{\sigma_s^2})$ | $(0, +\infty)$ |
| $\sigma_D^2$, if $\widehat{\sigma_s^2} < \widehat{\sigma_p^2}$ | $\mathrm{IG}(10, 0.1/(\widehat{\sigma_p^2} - \widehat{\sigma_s^2}))$ | $(0, +\infty)$ |
| $\sigma_\epsilon^2$ | $\mathrm{IG}(1, 100/\widehat{\sigma_s^2})$ | $(0, +\infty)$ |

We use a Metropolis Hastings algorithm to make draws from the posterior distribution of $\boldsymbol{\theta}_c$ given the data and a fixed $\boldsymbol{t}$.

- Let $\eta(\boldsymbol{x}) = E_\epsilon(Y^p(\boldsymbol{x}))$.
- Define the ideal value of $\boldsymbol{t}$

$$\boldsymbol{t}^\star = \arg\min_{\boldsymbol{t}} \int\int (\eta(\boldsymbol{x}) - y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}))^2 [\boldsymbol{\theta}_c | \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}] d\boldsymbol{\theta}_c d\boldsymbol{x}.$$

- Let $\eta(\boldsymbol{x}) = E_\epsilon(Y^p(\boldsymbol{x}))$.
- Define the ideal value of $\boldsymbol{t}$

$$\boldsymbol{t}^\star = \arg\min_{\boldsymbol{t}} \int\int (\eta(\boldsymbol{x}) - y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}))^2 [\boldsymbol{\theta}_c | \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}] d\boldsymbol{\theta}_c d\boldsymbol{x}.$$

We estimate $(\eta(\boldsymbol{x}) - y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}))^2 = \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ by its BLUP

$$E\left(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c\right) =$$
$$E_{[\boldsymbol{\phi} | \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c]}[E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\phi}, \boldsymbol{\theta}_c)],$$

where $\boldsymbol{\phi}$ denotes all the remaining model parameters.

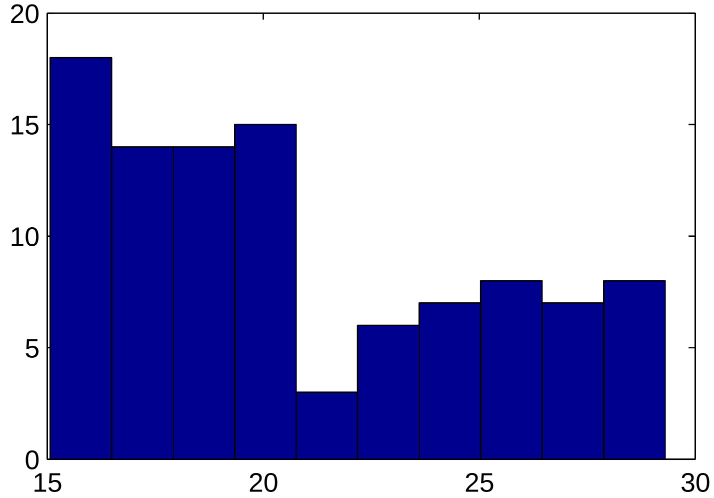
• We estimate $\boldsymbol{t}^\star$ by

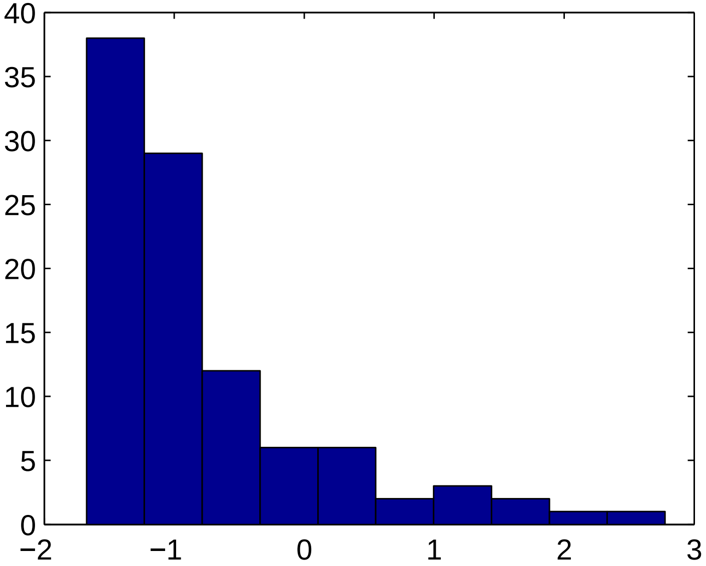
$$\begin{aligned}\widehat{\boldsymbol{t}^\star} &= \underset{\boldsymbol{t}}{\operatorname{argmin}} \int\int E(\Delta^2(\boldsymbol{x},\boldsymbol{\theta}_c,\boldsymbol{t})|\boldsymbol{y}^s,\boldsymbol{y}^p,\boldsymbol{t},\boldsymbol{\theta}_c)[\boldsymbol{\theta}_c|\boldsymbol{y}^s,\boldsymbol{y}^p,\boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x} \\ &= \underset{\boldsymbol{t}}{\operatorname{argmin}} \int E_{[\boldsymbol{\phi},\boldsymbol{\theta}_c|\boldsymbol{y}^s,\boldsymbol{y}^p,\boldsymbol{t}]}[E(\Delta^2(\boldsymbol{x},\boldsymbol{\theta}_c,\boldsymbol{t})|\boldsymbol{y}^s,\boldsymbol{y}^p,\boldsymbol{t},\boldsymbol{\phi},\boldsymbol{\theta}_c)]d\boldsymbol{x},\end{aligned}$$

where we approximate the latter by draws from an MCMC algorithm.

• Posterior of $\boldsymbol{\theta}_c$: $[\boldsymbol{\theta}_c \mid \boldsymbol{y}^p, \boldsymbol{y}^s, \widehat{\boldsymbol{t}^\star}]$.

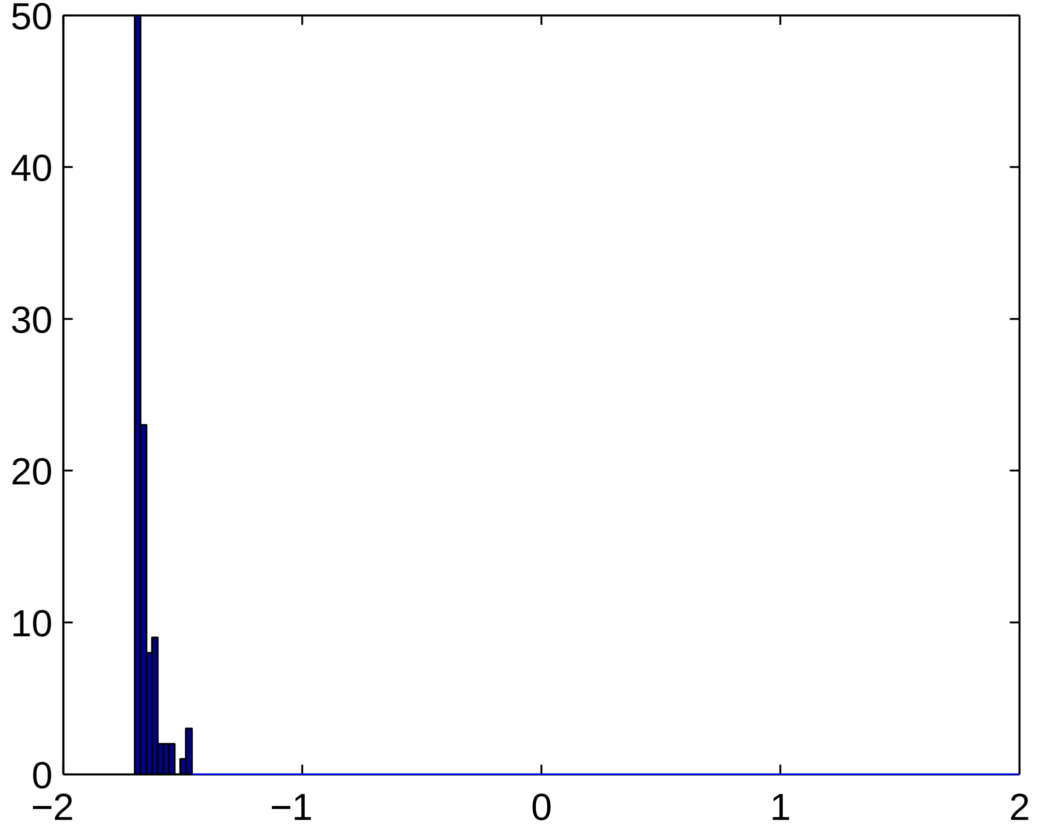
- **The result from Bayesian calibration program**




Simulated posterior distributions of **load discretization** (the left panel) and **initial position** (the right panel).

- **The result from our method**




Simulated posterior distribution of **initial position** with **estimated load discretization** $\widehat{t^{\star}} = 19$ .

- **Prediction** for $\eta(\boldsymbol{x})$

$$\widehat{\eta}(\boldsymbol{x}) \approx \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(Y^s(\boldsymbol{x}, \widehat{\boldsymbol{\theta}_{c,l}}, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \widehat{\boldsymbol{\theta}_{c,l}}, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}_{c,l}}, \widehat{\boldsymbol{\phi}_l}).$$

- **Predictive interval**

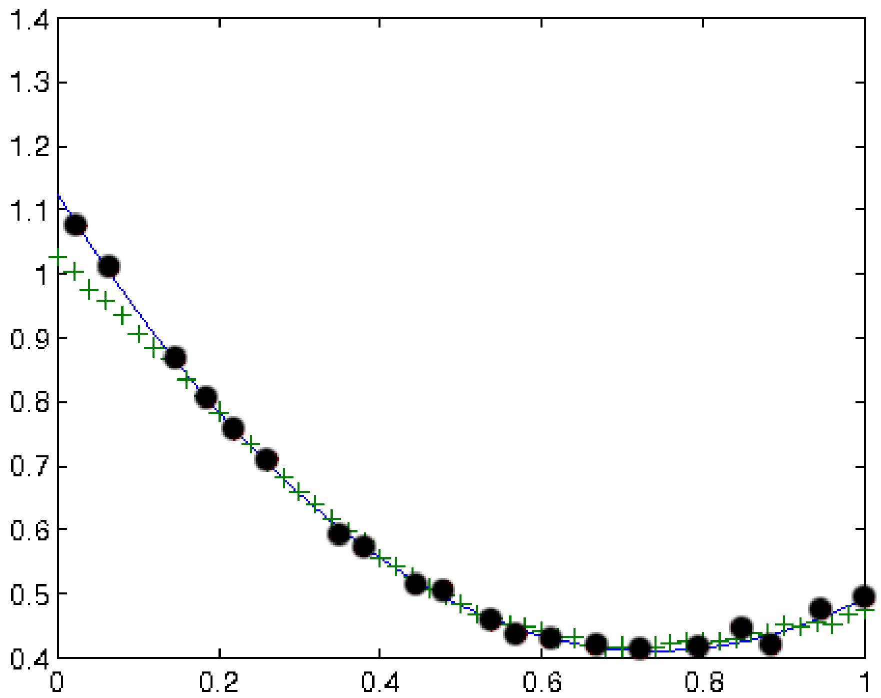
$$\widehat{\eta}(\boldsymbol{x}) \pm z^{\alpha/2} \sqrt{Var(Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star})},$$

where $z^{\alpha/2}$ is the upper $\alpha/2$ critical point of the standard normal distribution.

# Toy example

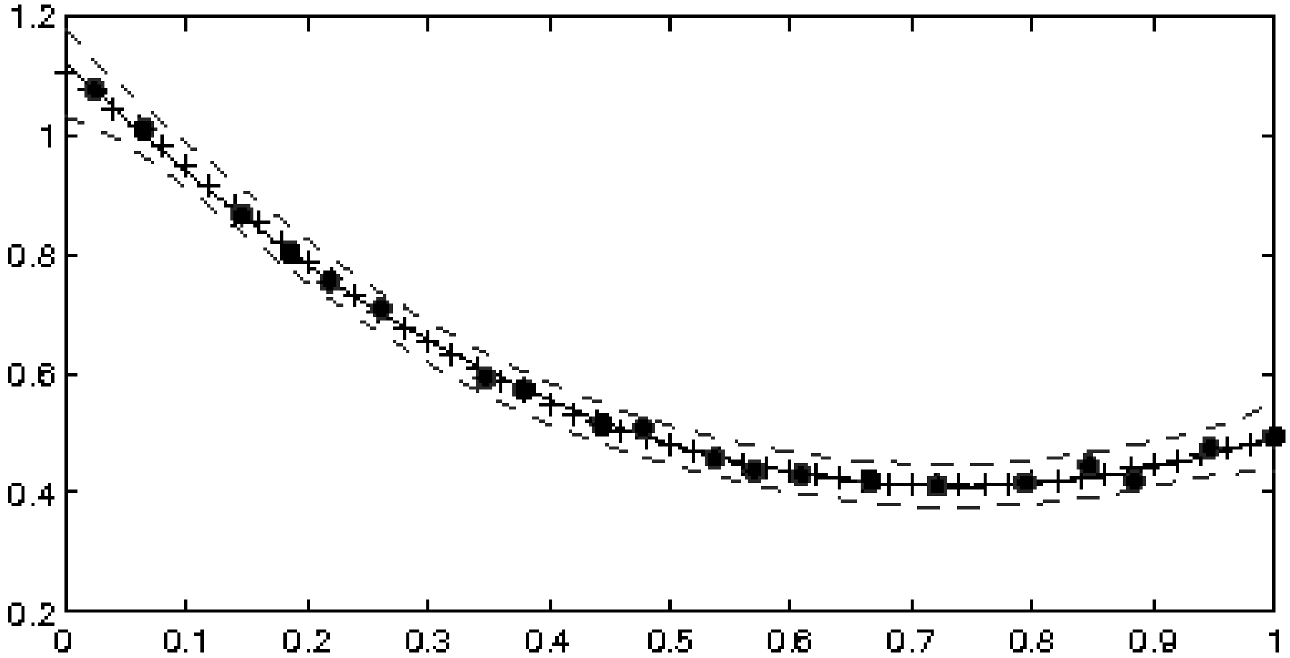
$$
\begin{aligned}
y^p(x) &= \exp(-x) + ((x-0.5)^2 - 0.125) + \mathbf{N}(0, 0.01^2) \\
y^s(x, (c_1, c_2), t) &= c_1 \times \exp(-c_2 \times x) + (t-0.5)^2 \times 10 \\
\{n_s, n_p, N_{MC}\} &= \{50, 20, 300\} \\
\eta(x) &= \exp(-x) + ((x-0.5)^2 - 0.125) \\
(\theta_{c_1}, \theta_{c_2}, t^\star) &= (1, 1, 0.5)
\end{aligned}
$$

The prediction error (RMSPE) over 101 points of $\eta(x)$ from Bayesian calibration: 0.1662.




The training data (solid circles), the true response curve (the solid line), and the predictions (pluses) obtained by the Bayesian calibration program.

The prediction error (RMSPE) of $\eta(x)$ from the Simultaneous Tuning and Calibration (STaC) program: 0.0445. Improvement rate $= 73.23\%$ $(= (0.1662 - 0.0445)/0.1662 \times 100\%)$.




The training data (solid circles), true response curve (the solid line), predictions (pluses), and 99% prediction bands (dashes) using the STaC program.

## III. Discussion and Conclusions

- STaC performs well for tuning, calibration, and prediction of the true input-output relationship.
- We recommend that tuning parameters should be set using a discrepancy measure.
- Future work
  - Speeding up the computation.
  - Estimating the uncertainty in $\widehat{\boldsymbol{t}^{\star}}$.
  - Extending the current method to other settings; e.g., multivariate outputs, quantitative and qualitative inputs.

**Thanks for your attention!**

# Appendix

Given $\boldsymbol{t}^{\star} = \text{argmin}_{\boldsymbol{t}} \int \int \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x}$ and $\widehat{\delta^2}(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) = E[\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c] =$ $E_{[\boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c]}[E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\phi}, \boldsymbol{\theta}_c)]$, we define

$$
\begin{aligned}
\widehat{\boldsymbol{t}^{\star}} &= \underset{\boldsymbol{t}}{\text{argmin}} \int \int E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c)[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x} \\
&= \underset{\boldsymbol{t}}{\text{argmin}} \int E_{[\boldsymbol{\phi}, \boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]}[E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\phi}, \boldsymbol{\theta}_c)]d\boldsymbol{x} \\
&\approx \underset{\boldsymbol{t}}{\text{argmin}} \int \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E[\Delta^2(\boldsymbol{x}, \widehat{\boldsymbol{\theta}_{c,l}}, \boldsymbol{t})|\widehat{\boldsymbol{\theta}_{c,l}}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{x} \\
&\approx \underset{\boldsymbol{t}}{\text{argmin}} \frac{1}{N_x} \frac{1}{N_{MC}} \sum_{i=1}^{N_x} \sum_{l=1}^{N_{MC}} E[\Delta^2(\boldsymbol{x}_i, \widehat{\boldsymbol{\theta}_{c,l}}, \boldsymbol{t})|\widehat{\boldsymbol{\theta}_{c,l}}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}].
\end{aligned}
$$

$$E[\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}] =$$
$$[E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi})]^2$$
$$+Var[\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}]$$

- $E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})\ |\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi})$ is a measure of the magnitude.
- $Var[\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\ \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}]$ is a measure of the variation.
- Our method favors a $\boldsymbol{t}$ that gives small, stable discrepancy between the computer code and the physical experiment.