

Reliability-Based Design Optimization with Model Bias and Data Uncertainty

Zhen Jiang and Wei Chen
Northwestern University

Yan Fu and Ren-Jye Yang
Ford Motor Company

ABSTRACT

Reliability-based design optimization (RBDO) has been widely used to obtain a reliable design via an existing CAE model considering the variations of input variables. However, most RBDO approaches do not consider the CAE model bias and uncertainty, which may largely affect the reliability assessment of the final design and result in risky design decisions. In this paper, the Gaussian Process Modeling (GPM) approach is applied to statistically correct the model discrepancy which is represented as a bias function, and to quantify model uncertainty based on collected data from either real tests or high-fidelity CAE simulations. After the corrected model is validated by extra sets of test data, it is integrated into the RBDO formulation to obtain a reliable solution that meets the overall reliability targets while considering both model and parameter uncertainties. The proposed technique is demonstrated through a vehicle design problem aiming at minimizing the vehicle weight through gauge optimization while satisfying reliability constraints. The RBDO result considering model uncertainty is compared with the one from conventional RBDO to demonstrate the benefits of the proposed method.

CITATION: Jiang, Z., Chen, W., Fu, Y. and Yang, R., "Reliability-Based Design Optimization with Model Bias and Data Uncertainty," *SAE Int. J. Mater. Manf.* 6(3):2013, doi:10.4271/2013-01-1384.

1. INTRODUCTION

Reliability-based design optimization (RBDO) is a frequently used approach in design under uncertainty aiming at obtaining a design solution that is most reliable under the variations of design variables. Beyond deterministic design optimization, which often drives the optimal design to the limit of design constraints and leaves little or no latitude for uncertainty, RBDO considers the uncertainty of controllable or uncontrollable variables associated with an existing CAE model. However, most of the conventional RBDO formulations do not account for the uncertainty from the CAE model; a CAE model is believed to be accurate enough for reliability assessment before it is implemented in RBDO process.

Unfortunately no model is perfect. A CAE model may be established based on the current knowledge of a physical process with approximations and simplifications; or it may directly come from response surface models (RSMs) that are fit to best match existing experimental data. For the former, the fidelity of the CAE model largely depends on how much the engineers know about the reality; the model parameters

may be in accord with past experiments, but there is no guarantee that it has strong predictive power for future designs if the underlying science is mis-captured. The inevitable underlying approximations and simplifications make the CAE model less reliable. The latter, a RSM, is even more problematic, where there is no physical meaning associated it. A RSM can describe the physical process accurately only at designs that have been experimentally tested or at those who are very close to such designs. The term *model uncertainty* is used to describe how much physical reality there is that remains unknown to the modelers; or, from an engineer's point of view, how trustworthy the current CAE model is to be used for further designs.

RBDO fails if the model is not accurate and is affected with large model uncertainty. Therefore, model bias correction is critical before implementing RBDO. Model bias correction is different from *model parameter tuning*, though, which modelers often use to adjust a CAE model to better match existing data. One can never achieve perfect match, however, between experiment data and a CAE model through

model parameter tuning. Yet model bias correction, which is more of a mathematical approach, deals with the discrepancy between model and experiment observations in a statistical way. There is a rich set of literature on what the possible sources of model uncertainty are and how they can be represented and/or mathematically quantified. Kennedy and O'Hagan 2001 [1] considered several different sources of uncertainty in model prediction: 1) *parameter uncertainty*, which is due to fixed but unknown parameters of a CAE model, e.g. damage or fracture coefficients, local free-fall acceleration, etc.; 2) *model bias* (or model discrepancy), which comes from missing underlying physics; 3) *interpolation uncertainty*, which is due to the lack of sufficient experimental tests and CAE simulations that in most of the cases are either economically or computationally expensive; 4) *numerical uncertainty*, to which is contributed mostly by numerical implementations of a CAE model, i.e. numerical integrations, infinite sum truncations, etc.; and finally 5) *experimental variability*, which represents the inevitable variability of experiments even when repeated many times using the exact same settings for model input variables. To quantify the model uncertainty, a general formulation is needed to take into consideration all these possible sources. The following general model uncertainty quantification formulation has been widely adopted in the previous work (Kennedy and O'Hagan 2001 [1]; Higdon, Kennedy et al. 2004 [2]; Bayarri, Berger et al. 2007 [3]; Liu, Bayarri et al. 2008 [4]; Arendt, Apley and Chen 2012 [5]):

$$y^e(\mathbf{x}) = y^m(\mathbf{x}, \boldsymbol{\theta}^*) + \delta(\mathbf{x}) + \varepsilon, \quad (1)$$

where $\mathbf{x}$ denotes model input variables, $\boldsymbol{\theta}^*$ denotes the true values of unknown model parameters $\boldsymbol{\theta}$, $y^e(\mathbf{x})$ denotes the experimental response as a function of variables $\mathbf{x}$, and $y^m(\mathbf{x}, \boldsymbol{\theta})$ denotes the CAE model response as a function of both design variables $\mathbf{x}$ and unknown parameters $\boldsymbol{\theta}$. $\delta(\mathbf{x})$ is a *bias function*, which mainly accounts for the model discrepancy. Finally, ε is a standalone term accounting for the experimental uncertainty, and it is assumed to be independent of $\mathbf{x}$ and $\boldsymbol{\theta}$. Numerical errors and interpolation uncertainties are considered for both $y^m(\mathbf{x}, \boldsymbol{\theta})$ and $\delta(\mathbf{x})$. This formulation is considered to be the most comprehensive that takes into account all the possible sources of uncertainty. With this formulation, the objectives of uncertainty quantification are to estimate the bias function $\delta(\mathbf{x})$ (aka *bias correction*) as well as to estimate the true values $\boldsymbol{\theta}^*$ of the unknown parameters, or a probability distribution of the possible values of $\boldsymbol{\theta}$ (aka *parameter calibration*).

In this paper, we focus on a degenerated model uncertainty quantification problem in which only model bias is considered. Basically there are no unknown parameters involved in the problem, with the formulation simplified as:

$$y^e(\mathbf{x}) = y^m(\mathbf{x}) + \delta(\mathbf{x}) + \varepsilon, \quad (2)$$

This is referred to as a problem of *model bias correction only*, and the goal is to estimate the bias function $\delta(\mathbf{x})$.

To quantify model uncertainty and to estimate the model bias, data are needed from both the CAE model simulations and real tests. In the situation when real tests are expensive, for example a vehicle crash experiment, test data may be collected from a high-fidelity CAE model to correct the bias of a coarse but cheap CAE model. Real experiments or simulations from a high-fidelity CAE model is called high-fidelity data source in this work. Figure 1 demonstrates the general model bias correction process. Simulations from a coarse CAE model are first collected, and then compared with high-fidelity data. Based on the difference, an estimation of the model bias can be obtained, with which the coarse CAE model can be "corrected". After that, a further validation is needed to ensure that the corrected CAE model is reliable; this validation step is done by comparing the corrected CAE model with some extra data from real tests or an equivalent high-fidelity CAE model. If the two match well, then the corrected CAE model is considered to be usable for future design (for example RBDO); otherwise, by either refining the original CAE model or collecting more data from either the simulation model and/or high-fidelity data sources, the model bias correction process is performed iteratively until the corrected model is trust-worthy, a conclusion drawn from the validation data set.

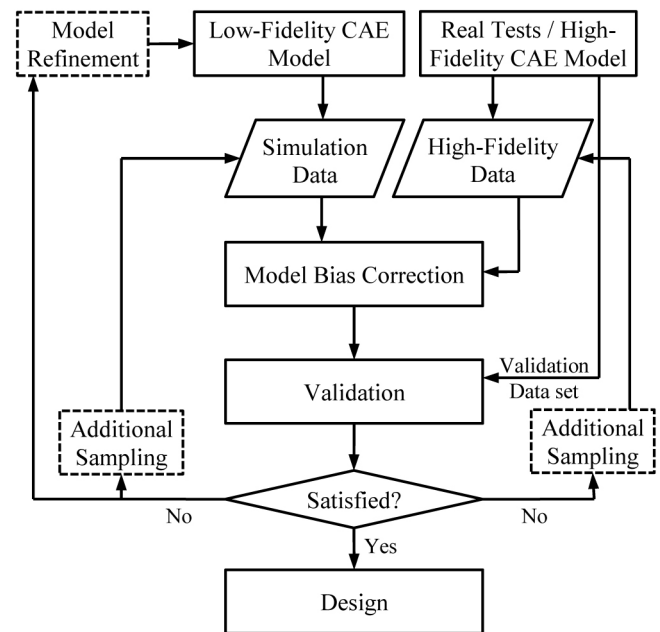


Figure 1. General model bias correction process

A key question that any possible methodology for bias correction needs to answer is: *how data from a low-fidelity CAE model should be compared to those from real tests / a high-fidelity CAE model?* The difficulty resides in three aspects: First, the two data sets most likely come from totally different settings of model input variables $\mathbf{x}$. In fact, there is no guarantee that the two sets provide an equal amount of

data; it is usually unaffordable to do so because of the high expense of collecting high-fidelity data. In other words, we cannot expect the data to be one-one comparable. Secondly, even if the two data sets have the same amount of data under same settings of input variables, it still requires efforts towards quantifying the model uncertainty at input settings that have not yet been tested. Thirdly, due to the experimental variability, experimentalists often conduct repeated tests for a specific input setting. How to incorporate information from repeated tests into the bias correction process remains a question. A side question is then how to incorporate the model uncertainty information into the RBDO process, where conventional approaches like First Order Reliability Method (FORM) [6] and Second Order Reliability Method (SORM) [7] are only applicable for assessing reliability under parameter uncertainty.

Much effort has been devoted for the model bias correction problem. Hasselman, Yap et al. 2005 [8] discussed the solution to a one-one comparable situation. Chen, Xiong et al. 2008 [9], Qian and Wu 2008 [10], Wang, Tsui et al. 2009 [11] introduced a Bayesian approach for bias correction and derived a Bayesian closed form solution for the situation when there is lack of data. For a general case with repeated test availability, a specific modeling technique is needed that can capture the overall characteristics of the CAE models and the real tests as well, so that they can be compared not directly on the data collection level, but on the modeling level.

To address the above needs, in this paper, we propose an efficient approach for model bias correction using the Gaussian process modeling technique, which is described in detail in Section 2. An immediate example of improving CAE models for vehicle frontal impact is provided to demonstrate the effectiveness of the approach. Section 3 then further extends the corrected model for design optimization under uncertainty, using a modified Monte Carlo Sampling approach for solving the RBDO formulation with consideration of both model and parameter uncertainties. Conclusions are made in Section 4.

2. GAUSSIAN PROCESS MODELING FOR MODEL BIAS CORRECTION

2.1. Basic Concepts of Gaussian Process

In probability theory, a *random process* (aka a *stochastic process*) is a collection of random variables associated with time or space. It is often used to describe the evolution of random variables over time and/or space, and the value at any specific location of the random process is random. For example, the number of customers at a shopping store can be modeled as a random process; it not only changes over time in a day, but at the exactly same time in different days it is also random and different.

A *Gaussian random process* (or in short, *Gaussian process*) $f(\bullet)$ is a random process that has two Gaussian properties. It can be represented in the following form:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), V(\mathbf{x}, \mathbf{x}')), \quad (3)$$

where $\mathbf{x}$ denotes time or multidimensional space location. $m(\mathbf{x})$ is the *mean function* of the Gaussian process, and $V(\mathbf{x}, \mathbf{x}')$ is the *covariance function* of the Gaussian process, representing the spatial covariance between two points ($\mathbf{x}$ and $\mathbf{x}'$) at the process. The two Gaussian properties are that 1) the random variable $f(\mathbf{x})$ at location $\mathbf{x}$ follows a normal distribution

$$f(\mathbf{x}) \sim N(m(\mathbf{x}), V(\mathbf{x}, \mathbf{x})), \quad (4)$$

and that 2) a finite collection of random variables $f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n)$ at locations $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ follows a multivariate normal distribution

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \sim N \left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} V(\mathbf{x}_1, \mathbf{x}_1) & \cdots & V(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ V(\mathbf{x}_n, \mathbf{x}_1) & \cdots & V(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \right). \quad (5)$$

A special Gaussian random process that has been widely used for metamodeling (meaning creating a surrogate model of an expensive model, called “modeling a model”) is a *stationary* Gaussian process, whose covariance function doesn't change when shifted in $\mathbf{x}$, namely $V(\mathbf{x}, \mathbf{x}') = V(\mathbf{x} + \Delta\mathbf{x}, \mathbf{x}' + \Delta\mathbf{x})$. A frequently used form of a stationary Gaussian process is:

$$f(\mathbf{x}) \sim GP(\mathbf{h}(\mathbf{x})^T \boldsymbol{\beta}, \sigma^2 \exp\left\{-\sum_{k=1}^p \omega_k (x_k - x'_k)^2\right\}) \quad (6)$$

where p is the dimension of $\mathbf{x}$, i.e. $\mathbf{x} = [x_1, x_2, \dots, x_p]^T$. $\mathbf{h}(\mathbf{x})$ is the vector of a user-predefined polynomial functions used to represent the mean, e.g. for a constant mean situation $\mathbf{h}(\mathbf{x}) = 1$, or for a linear mean situation $\mathbf{h}(\mathbf{x}) = [1, x_1, x_2, \dots, x_p]^T$. $\boldsymbol{\beta}$ is the vector of coefficients associated with $\mathbf{h}(\mathbf{x})$ for polynomial regression of the mean, σ is the standard deviation for a single random variable in the random process, and $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_p]^T$ is the vector of roughness parameters that are used to capture the nonlinearity of the process. Other forms of a stationary Gaussian process can be found in literature [12] for different purposes.

In metamodeling, Gaussian process approach is applied for response surface modeling. From a set $\mathbf{d}$ of existing data, the approach estimates $\boldsymbol{\beta}$, σ and $\boldsymbol{\omega}$, which are called *hyperparameters* of the Gaussian process, by the Maximum Likelihood Estimation (MLE) approach to maximize the *likelihood function*

$$\max_{\boldsymbol{\beta}, \sigma, \boldsymbol{\omega}} L(\boldsymbol{\beta}, \sigma, \boldsymbol{\omega}; \mathbf{d}) = p(\mathbf{d}; \boldsymbol{\beta}, \sigma, \boldsymbol{\omega}), \quad (7)$$

where $p(\bullet)$ denotes the probability density function (*pdf*). The likelihood function basically serves as a judgment on how likely the hyperparameters $\boldsymbol{\beta}$, σ and $\boldsymbol{\omega}$ are to be certain values given the data $\mathbf{d}$ we know. After the most likely values of hyperparameters are determined, the Gaussian process is fully determined and can be used to predict the values at other different locations, and the covariance function is used to calculate the prediction variance at each location, in which way it quantifies the model uncertainty. Due to its flexibility and ability to capture the nonlinearity of the underlying model through different settings of hyperparameters, the Gaussian process approach remains one of the most popular approaches in metamodeling to date, and is believed to be applicable for a large variety of different models. The well-known Kriging method [13] is considered to be a special case of the Gaussian process approach.

2.2. Gaussian Process Modeling for CAE Model and Bias Function

Here we borrow the basic concept of Gaussian process modeling for model bias correction. Going back to the general model bias correction formulation (2), assume that the CAE model $y^m(\mathbf{x})$ can be replaced by a Gaussian process with hyperparameters $\boldsymbol{\beta}^m$, σ_m and $\boldsymbol{\omega}^m$

$$y^m(\mathbf{x}) \sim GP\left(\mathbf{h}^m(\mathbf{x})^T \boldsymbol{\beta}^m, \sigma_m^2 \exp\left\{-\sum_{k=1}^p \omega_k^m (x_k - x'_k)^2\right\}\right), \quad (8)$$

and that the bias function $\delta(\mathbf{x})$ can be modeled as a Gaussian process with hyperparameters $\boldsymbol{\beta}^\delta$, σ_δ and $\boldsymbol{\omega}^\delta$

$$\delta(\mathbf{x}) \sim GP\left(\mathbf{h}^\delta(\mathbf{x})^T \boldsymbol{\beta}^\delta, \sigma_\delta^2 \exp\left\{-\sum_{k=1}^p \omega_k^\delta (x_k - x'_k)^2\right\}\right), \quad (9)$$

though at this stage all those hyperparameters ($\boldsymbol{\beta}^m$, σ_m , $\boldsymbol{\omega}^m$, $\boldsymbol{\beta}^\delta$, σ_δ , $\boldsymbol{\omega}^\delta$) remain unknown to us. Furthermore we assume that the experimental variability ε follows a zero-mean normal distribution $\varepsilon \sim N(0, \lambda)$, and then additively the experimental response $y^e(\mathbf{x})$ can also be replaced by a Gaussian process:

$$\begin{aligned} y^e(\mathbf{x}) \sim GP & \left(\mathbf{h}^m(\mathbf{x})^T \boldsymbol{\beta}^m + \mathbf{h}^\delta(\mathbf{x})^T \boldsymbol{\beta}^\delta, \right. \\ & \sigma_m^2 \exp\left\{-\sum_{k=1}^p \omega_k^m (x_k - x'_k)^2\right\} \\ & \left. + \sigma_\delta^2 \exp\left\{-\sum_{k=1}^p \omega_k^\delta (x_k - x'_k)^2\right\} + \lambda I\{\mathbf{x} = \mathbf{x}'\} \right), \end{aligned} \quad (10)$$

where I is the indicator function, equaling 1 when $\mathbf{x} = \mathbf{x}'$ holds and equaling 0 otherwise. Note that the above summation is true *if and only if* the CAE model $y^m(\mathbf{x})$, the bias function $\delta(\mathbf{x})$ and the experimental variability ε are independent of

each other. Otherwise an extra step is needed to eliminate the correlation between them (as much as possible) before the model bias correction process. There is no standard way for the dependency elimination; nevertheless, for certain simple cases it would not be difficult. For example, if a strong linear relationship is observed between $y^m(\mathbf{x})$ and $\delta(\mathbf{x}) + \varepsilon$, a linear regression can be performed between $y^m(\mathbf{x})$ and $\delta(\mathbf{x}) + \varepsilon$, based on which $y^m(\mathbf{x})$ and $\varepsilon(\mathbf{x})$ can both be transformed so that they don't have much correlation. This issue will be revisited in the case study in Section 2.4, while for the time being it is assumed that the additive relation in Equation (10) is applicable.

With the assumption of independency between $y^m(\mathbf{x})$, $\delta(\mathbf{x})$ and ε , the covariance of CAE model response $y^m(\mathbf{x})$ and the experimental response $y^e(\mathbf{x})$ can be found as:

$$\begin{aligned} & \text{cov}(y^e(\mathbf{x}), y^m(\mathbf{x}')) \\ &= \text{cov}(y^m(\mathbf{x}) + \delta(\mathbf{x}) + \varepsilon, y^m(\mathbf{x}')) \\ &= \text{cov}(y^m(\mathbf{x}), y^m(\mathbf{x}')) + \text{cov}(\delta(\mathbf{x}), y^m(\mathbf{x}')) + \text{cov}(\varepsilon, y^m(\mathbf{x}')) \\ &= V^m(\mathbf{x}, \mathbf{x}'), \end{aligned} \quad (11)$$

where $V^m(\mathbf{x}, \mathbf{x}')$ is the covariance function of CAE model in Equation (8).

Suppose that a set of M data points

$$\mathbf{d}^m = [y^m(\mathbf{x}_1^m), y^m(\mathbf{x}_2^m), \dots, y^m(\mathbf{x}_M^m)]^T \quad (12)$$

has been collected at M input settings $\mathbf{X}^m = [\mathbf{x}_1^m, \mathbf{x}_2^m, \dots, \mathbf{x}_M^m]^T$ from the CAE model simulations. Similarly suppose that a set of N data points

$$\mathbf{d}^e = [y^e(\mathbf{x}_1^e), y^e(\mathbf{x}_2^e), \dots, y^e(\mathbf{x}_N^e)]^T \quad (13)$$

has been collected at N input settings $\mathbf{X}^e = [\mathbf{x}_1^e, \mathbf{x}_2^e, \dots, \mathbf{x}_N^e]^T$ from the real tests or a high-fidelity CAE simulator. The combination of Equations (8), (10) and (11) yields that

$$\begin{aligned} \begin{bmatrix} \mathbf{d}^m \\ \mathbf{d}^e \end{bmatrix} \sim N & \left(\begin{bmatrix} \mathbf{H}^m(\mathbf{X}^m) & \mathbf{0} \\ \mathbf{H}^m(\mathbf{X}^e) & \mathbf{H}^\delta(\mathbf{X}^e) \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}^m \\ \boldsymbol{\beta}^\delta \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \mathbf{V}^m(\mathbf{X}^m, \mathbf{X}^m) & \mathbf{V}^m(\mathbf{X}^e, \mathbf{X}^m)^T \\ \mathbf{V}^m(\mathbf{X}^e, \mathbf{X}^m) & \mathbf{V}^m(\mathbf{X}^e, \mathbf{X}^e) + \mathbf{V}^\delta(\mathbf{X}^e, \mathbf{X}^e) + \lambda \mathbf{I} \end{bmatrix} \right), \end{aligned} \quad (14)$$

where

$$\begin{aligned}\mathbf{H}^m(\mathbf{X}) &= [\mathbf{h}^m(\mathbf{x}_1), \mathbf{h}^m(\mathbf{x}_2), \dots, \mathbf{h}^m(\mathbf{x}_K)]^T, \\ \mathbf{H}^\delta(\mathbf{X}) &= [\mathbf{h}^\delta(\mathbf{x}_1), \mathbf{h}^\delta(\mathbf{x}_2), \dots, \mathbf{h}^\delta(\mathbf{x}_K)]^T, \\ \mathbf{V}^m(\mathbf{X}, \mathbf{X}') &= \begin{bmatrix} V^m(\mathbf{x}_1, \mathbf{x}'_1) & \cdots & V^m(\mathbf{x}_1, \mathbf{x}'_{K'}) \\ \vdots & \ddots & \vdots \\ V^m(\mathbf{x}_K, \mathbf{x}'_1) & \cdots & V^m(\mathbf{x}_K, \mathbf{x}'_{K'}) \end{bmatrix}, \\ \mathbf{V}^\delta(\mathbf{X}, \mathbf{X}') &= \begin{bmatrix} V^\delta(\mathbf{x}_1, \mathbf{x}'_1) & \cdots & V^\delta(\mathbf{x}_1, \mathbf{x}'_{K'}) \\ \vdots & \ddots & \vdots \\ V^\delta(\mathbf{x}_K, \mathbf{x}'_1) & \cdots & V^\delta(\mathbf{x}_K, \mathbf{x}'_{K'}) \end{bmatrix}.\end{aligned}$$

$\mathbf{I}$ is an identity matrix. K and K' are the maximum indices for two matrices $\mathbf{X}$ and $\mathbf{X}'$, respectively; they are equal to M for $\mathbf{X}^m$ and N for $\mathbf{X}^e$. For simplicity, Equation (14) is rewritten in the following concise form:

$$\mathbf{d} \sim N(\mathbf{H}\boldsymbol{\beta}, \mathbf{V}_d). \quad (15)$$

The next step is to carry out the estimates for all the unknown hyperparameters $\boldsymbol{\beta}^m$, σ_m , ω^m , $\boldsymbol{\beta}^\delta$, σ_δ , and ω^δ in Equations (8, 9, 10).

Having been briefly mentioned in Equation (7), the Maximum Likelihood Estimation (MLE) is a useful tool for estimating the hyperparameters based on existing data set $\mathbf{d}^m$ and $\mathbf{d}^e$. The likelihood function of the hyperparameters given the data set $\mathbf{d}=[\mathbf{d}^m; \mathbf{d}^e]$ is

$$\begin{aligned}L(\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda; \mathbf{d}) \\ = p(\mathbf{d}; \boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda) \\ \propto |\mathbf{V}_d|^{-1/2} |\mathbf{W}|^{1/2} \exp\left\{-\frac{1}{2}(\mathbf{d}-\mathbf{H}\boldsymbol{\beta})^T \mathbf{V}_d^{-1}(\mathbf{d}-\mathbf{H}\boldsymbol{\beta})\right\},\end{aligned} \quad (16)$$

where $\mathbf{W}=(\mathbf{H}^T \mathbf{V}_d^{-1} \mathbf{H})^{-1}$. To solve the MLE problem

$$\max_{\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda} L(\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda; \mathbf{d}), \quad (17)$$

it is equivalent to maximize the *log-likelihood function*

$$\max_{\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda} \ell(\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda; \mathbf{d}), \quad (18)$$

where

$$\begin{aligned}\ell(\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda; \mathbf{d}) \\ = \log L(\boldsymbol{\beta}^m, \sigma_m, \omega^m, \boldsymbol{\beta}^\delta, \sigma_\delta, \omega^\delta, \lambda; \mathbf{d}) \\ = -\frac{1}{2} \log |\mathbf{V}_d| + \frac{1}{2} \log |\mathbf{W}| - \frac{1}{2}(\mathbf{d}-\mathbf{H}\boldsymbol{\beta})^T \mathbf{V}_d^{-1}(\mathbf{d}-\mathbf{H}\boldsymbol{\beta}) + c.\end{aligned} \quad (19)$$

c is some trivial constant quantity. According to the first-order optimality condition,

$$\mathbf{0} = \frac{\partial \ell(\cdot)}{\partial \boldsymbol{\beta}} \bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} = (\mathbf{d}-\mathbf{H}\hat{\boldsymbol{\beta}})^T \mathbf{V}_d^{-1} \mathbf{H} = \mathbf{d}^T \mathbf{V}_d^{-1} \mathbf{H} - \hat{\boldsymbol{\beta}}^T \mathbf{H}^T \mathbf{V}_d^{-1} \mathbf{H},$$

which yields that the estimate of $\boldsymbol{\beta}$ is related to the estimates of other hyperparameters following

$$\hat{\boldsymbol{\beta}} = \mathbf{W} \mathbf{H}^T \mathbf{V}_d^{-1} \mathbf{d}. \quad (20)$$

Based on Equations (18, 19, 20), the MLE problem that needs to be solved is

$$\max_{\sigma_m, \omega^m, \sigma_\delta, \omega^\delta, \lambda} -\frac{1}{2} \log |\mathbf{V}_d| + \frac{1}{2} \log |\mathbf{W}| - \frac{1}{2}(\mathbf{d}-\mathbf{H}\hat{\boldsymbol{\beta}})^T \mathbf{V}_d^{-1}(\mathbf{d}-\mathbf{H}\hat{\boldsymbol{\beta}}). \quad (21)$$

There are, however, no closed solutions for the estimates of hyperparameters in Equation (21), which calls for a numerical optimization algorithm. Due to the highly nonlinear behavior of the MLE objective, genetic algorithm (GA) and simulated annealing (SA) are among those algorithms that can be widely used to solve MLE.

2.3. Gaussian Process Modeling for Prediction After Model Bias Correction

With certain optimization schemes, the MLE problem can be solved and hence the Gaussian processes for CAE model $y^m(\mathbf{x})$, bias function $\delta(\mathbf{x})$, and the experimental response $y^e(\mathbf{x})$ are determined. Subsequently the Gaussian processes are used for prediction at untested input settings (new design points) to answer the following two questions: 1) *how much the bias would be at the untested designs*, and 2) *what values the experimental response, which is the true response associated with experimental variability, is likely to take at the untested designs*.

Denote the untested designs of interest as $\mathbf{X}^p$ that consists of a series of L designs $\mathbf{X}^p = [\mathbf{x}_1^p, \mathbf{x}_2^p, \dots, \mathbf{x}_L^p]^T$. Following the same routine as in Equation (11), we can derive the covariance between $y^e(\mathbf{x}^p)$ and $y^e(\mathbf{x})$ as well as between $y^e(\mathbf{x}^p)$ themselves:

$$\text{cov}(y^e(\mathbf{x}^p), y^e(\mathbf{x})) = V^m(\mathbf{x}^p, \mathbf{x}) + V^\delta(\mathbf{x}^p, \mathbf{x}), \quad (22)$$

$$\text{cov}(y^e(\mathbf{x}^p), y^e(\mathbf{x}^p)) = V^m(\mathbf{x}^p, \mathbf{x}^p) + V^\delta(\mathbf{x}^p, \mathbf{x}^p) + \lambda. \quad (23)$$

Hence similarly to Equations (14~15), we obtain that