A Response-Modeling Alternative to Surrogate Models for Support in Computational Analyses

Brian Rutherford

Sandia National Laboratories, P.O. Box 5800, MS 0829, Albuquerque, NM 87185, USA
E-mail: bmruthe@sandia.gov

Abstract: Often, the objectives in a computational analysis involve characterization of system performance based on some function of the computer response. In general, this characterization includes (at least) an estimate or prediction for the performance measure and an estimate of the associated uncertainty. Surrogate models can be used to approximate the response in regions where simulations were not performed. Most surrogate modeling approaches, however, are based on smoothing and uncertainty in the response is typically specified in a point-wise (in the input space) fashion. Together these aspects of the surrogate model construction might limit their capabilities.

One alternative is to construct a probability measure, $G(\mathbf{r})$ for the computer response, $\mathbf{r}$, based on available data. This "response-modeling" approach will permit probability estimation for an arbitrary event, $E(\mathbf{r})$, based on the computer response. In this general setting: $prob(E) = \int_r I(E(\mathbf{r}))dG(\mathbf{r})$ where I is an indicator function. Furthermore, one can use $G(\mathbf{r})$ to calculate an induced distribution on the performance measure, pm. For prediction problems where the performance measure is a scalar, the performance measure distribution F_{pm} is determined by: $F_{pm}(z) = \int_r I(pm(\mathbf{r}) \le z)dG(\mathbf{r})$. We introduce response models for scalar computer output and then generalize the approach to more complicated responses that utilize multiple response models.

Keywords: computational simulation, experimental design, meta-model, prediction, reliability, response-modeling, response surface, surrogate models.

1. INTRODUCTION

Enhanced software methodology and improved computing hardware have advanced the state of simulation technology to a point where large physics-based codes can be a major contributor in many systems analyses. This shift toward the use of computational methods has brought with it new research challenges in a number of areas including model validation, (model-based) prediction and characterization of input, modeling and predictive uncertainty. It is these challenges that have motivated the work described in this paper.

The problem considered here is one of characterizing system performance based on results of a computer model. It is assumed that the model is expensive to run and consequently only a limited number of evaluations can be performed. It is assumed, further, that the model has been validated and hence the model has been determined to provide adequate results for the present application. For simplicity of presentation, we assume the model produces a single response for a given set of inputs. Different responses depending on uncertain model parameters (calibration parameters) can, however, be accommodated through the methods discussed here. The tough issues of model validation and calibration are not addressed here.

We consider a number of specific system performance measures for illustration -- others are possible. These performance measures are based on the model responses that may be scalars or may be vectors or functions. In the introduction to the response models, we work with a scalar computer response. A scalar response is modeled by one response model. In the remainder of the paper, we generalize the approach using a functional response over time. The later applications illustrate methodology for using multiple response models to accommodate more general computer model outputs.

The response model described here is an atomic probability measure for the response calculated using a limited number of computational results. We use this measure to approximate probabilities associated with arbitrary events that are based on the response, although the major objective is to obtain a probability distribution function for the performance measure. We provide details on how the response models can be constructed and how they can be used in fairly general applications.

In this introductory section, we: (1.1) review some of the possible objectives in computational modeling; (1.2) discuss how surrogate models can assist in addressing these objectives, review one specific parametric form for surrogate models and mention some possible limitations in their use; and (1.3) discuss an alternative response-modeling strategy that overcomes some of these limitations. In the remainder of this paper, Section 2 provides a more detailed account of the response-modeling approach. A simple example is introduced that illustrates the construction of the measure G(**r**) for a scalar computer response. Section 3 extends the approach to more general types of computer responses providing two examples where multiple response models are used to characterize a functional output in time.

1.1. Objectives in Computational Modeling

Often, the objectives in a computational analysis involve the characterization of system performance based on some function of the response, **r**. We consider applications where $\mathbf{r}(\mathbf{x})$ is the computer response depending on the p-dimensional input **x.** The inputs may or my not be modeled probabilistically (with distribution $F(\mathbf{x})$). For given **r**, some common performance measures may be computed as:

$pm(\mathbf{r}) = \mathbf{r}(\mathbf{x}^*)$ (simple performance prediction problem);

$pm(\mathbf{r}) = \int \mathbf{r}(\mathbf{x})dF(\mathbf{x})$ (average performance prediction problem);

$pm(\mathbf{r}) = \min_{\mathbf{x}} \mathbf{r}(\mathbf{x})$ (worst-case performance prediction problem);

$pm(\mathbf{r}) = \{\mathbf{x}^* : \mathbf{r}(\mathbf{x}^*) = \min_{\mathbf{x}}(\mathbf{r}(\mathbf{x}))\}$ (engineering design or optimization problem); or

$pm(\mathbf{r}) = 1 - \int I(\mathbf{r}(\mathbf{x}) \varepsilon R^*) dF(\mathbf{x})$ (reliability prediction problem)

where, R^*, the failure region, is some subset of the response space and I is an indicator function taking on the value 1 when the enclosed expression is true. In general, we desire both a prediction for the performance measure and an estimate of prediction uncertainty.

1.2. Surrogate Modeling for Computational Analyses

1.2.1. Surrogate models in computational analysis

Surrogate models can perform a number of functions in support of a computational analysis. Through interpolation extrapolation and/or integration, these models can be used to address complex problems involving experimental design, system analysis and prediction.

1.2.2. A common surrogate model form

One model that is commonly used as a surrogate response is the Gaussian Process model $\eta(\mathbf{x})$. A general form of the model is specified through its mean function $m(\mathbf{x})$ and spatial covariance function $\sigma^2\mathbf{C}(\mathbf{x},\mathbf{x}')$ for input vectors $\mathbf{x}$ and $\mathbf{x}'$. This model has been used successfully in numerous engineering applications, see [1] – [3] for typical assumptions and restrictions on the forms of m and $\mathbf{C}$.

1.2.3. Possible limitations to surrogate modeling

Depending on the performance measure, the smoothing involved in surrogate model construction and method of uncertainty characterization of the surrogate model might lead to difficulties. Consider the worst-case performance measure where $pm(\mathbf{r}) = \min_{\mathbf{x}} \mathbf{r}(\mathbf{x})$. The performance measure estimate based in the surrogate model will not give an accurate estimate of the minimum response value. Even an expected value for the worst-case response would be difficult to approximate using the information retained. Figure 1(a) illustrates the difficulty. Without information on the relationships of points in the lower regions of the curve, the probabilities are difficult to determine. Similarly, for the reliability performance measure, $pm(\mathbf{r}) = 1 - \int I(\mathbf{r}(\mathbf{x})\varepsilon R^*)dF(\mathbf{x})$, while the surrogate model estimate might provide a reasonable estimate of reliability, the needed information is not retained in the surrogate model construct to accurately quantify uncertainty in this estimate.

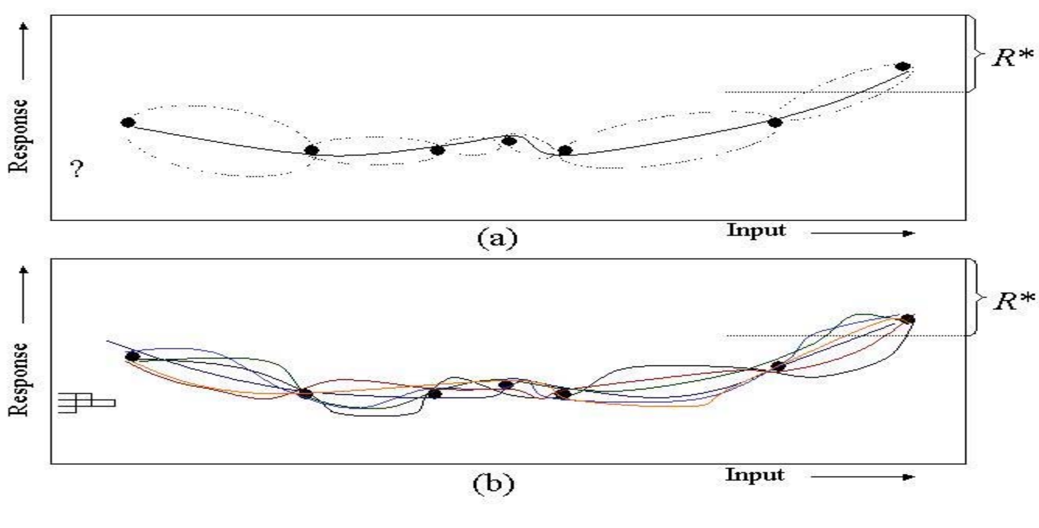

Figure 1. Hypothetical models for the response in a 1-input problem: (a) a surrogate model estimate with point-wise uncertainty bounds; and (b) a response-model with individual “realizations” forming an atomic measure over the response space. The histogram on the left indicates how a distribution might be constructed for a worst-case performance measure.

1.3. An Alternative Model for the Response

One alternative characterization for the system response is illustrated in Figure 1b and discussed in detail in Section 2 below. This "response-modeling" approach can avoid the problems discussed in the previous subsection. It consists of constructing an atomic measure over the response space that is based on assumptions concerning the appropriate model form and on the available computer response data. We refer to elements of the measure as "realizations" and assign them equal probability.

Once this measure has been established for the response space, we can approximate the probability associated with any event $E(\mathbf{r})$ through $prob(E) = \int_r I(E(\mathbf{r}))dG(\mathbf{r})$ and for the performance measure distribution through $F_{pm}(z) = \int_r I(pm(\mathbf{r}) \leq z)dG(\mathbf{r})$. In Figure 1b we illustrate how this measure might be used on that 1-input problem. The histogram on the left of the figure provides an approximation to the density function for the worst-case performance measure -- a value that was difficult to estimate using the surrogate model. Similarly, we could demonstrate the uncertainty related to the reliability prediction by drawing a histogram of the reliability values computed using the individual realizations.

2. RESPONSE MODELING

2.1. Modeling Objectives

We construct a response-model as a discrete ensemble of realizations that could be interpreted as "probable" descriptions of the computer response as a function of the computer inputs. The realizations are constructed in the spirit of a Latin Hypercube sample [4] where they are generated to span the uncertainty range of the response while attempting to satisfy the consistency property stated in the next paragraph. The ensemble is used to approximate an uncertainty distribution for the fixed but unknown true response surface that captures the uncertainty in the response resulting from the knowledge being based on a limited number, n system evaluations. Formally, the response ensemble consists of a set of k realizations: $\mathbf{R} = (r^i(\mathbf{x}), i = 1,...,k, \forall \mathbf{x})$; G($\mathbf{r}$) assigns a probability $1/k$ to each realization $r^i(\mathbf{x})$. Using this formulation, the expressions above become: $prob(E) = 1/k \sum_i I(E(r^i))$ and $F_{pm}(z) = 1/k \sum_i I(pm(r^i) \leq z)$.

Ideally, $\mathbf{R}$ is constructed in a manner consistent with the data $\mathbf{y} = y(\mathbf{x}_i)$, i=1,…,$n$ in the following sense: for any given event E based on the response, if the conditional probability $P(E \mid \mathbf{y}) = p$ then the expected number of realizations $r^i(\mathbf{x})$ satisfying E, (the expectation taken over repeated application of the response-modeling process) is kp.

The assertion $P(E \mid \mathbf{y}) = p$ requires assumptions for mathematical formalization. The response model is based on these assumptions addressing functional form and appropriate methods of construction. Decisions regarding the modeling assumptions are necessarily somewhat arbitrary. Some of the issues are addressed in [5] – [6], others are the topic of our current research. Some of the possible inaccuracies resulting from the assumptions tend to cancel each other out when making relative evaluations like comparing experimental design alternatives (our primary application of the response models). Figure 2 shows an example response that is used throughout this section. The response is an analytical function that is simple to evaluate, but is used here, for illustration, in place of the expensive computational simulation model. A twenty-five-realization response model is constructed based on fifteen functional evaluations as indicated by the stars in Figure 2b. In the next subsection, we provide the details for constructing the response model.

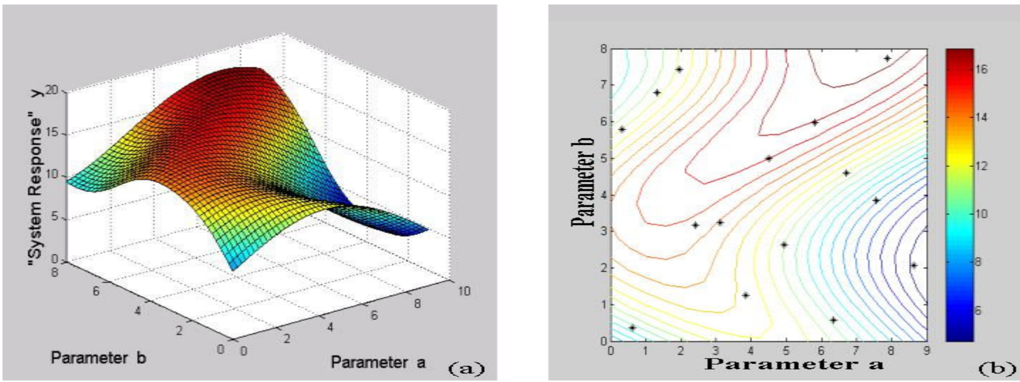

Figure 2. Analytical example response surface and contours based on two inputs.

2.2. Response Model Construction

Listed next are the steps used in construction of the realizations for the examples given in this paper. We utilize the Gaussian process model referenced in Subsection 1.2.2. We consider m to be a low order polynomial in **x** and restrict **C** to be of the form:

$$\mathbf{C}(\mathbf{x},\mathbf{x}') = \prod_{i=1}^{p} C\left(\left|x_i - x_i'\right|\right) = \prod_{i=1}^{p} e^{-\phi_i \left\|\mathbf{x}_i - \mathbf{x}_i'\right\|} \text{ for any } \mathbf{x} \text{ and } \mathbf{x}'$$

where $\|\cdot\|$ is the Euclidian norm and the ϕ_i are estimated from the data. In our case, **C** is the covariance for the residual response after fitting the low order polynomial as described in more detail below. It is more convenient for our application to decompose the Gaussian process model and use the form $\eta(\mathbf{x}) = P(\mathbf{x}) + \varepsilon(\mathbf{x})$ where $P(\mathbf{x})$ is a polynomial in **x** and $\varepsilon(\mathbf{x})$ is a zero-mean Gaussian process model. Steps (1) through (3) below describe construction of the polynomial component; steps (4), (5) and (6) describe construction of the Gaussian process term; step (7) combines the two elements.

1) Evaluate main effects, quadratic terms, and interactions, where possible, using the initial data, settling on an appropriate polynomial regression model.
2) Estimate the regression coefficients and their covariance structure.
3) Generate k sets of coefficients (assuming a multivariate normal for their joint distribution) using a Latin Hypercube design with the appropriate correlation structure imposed on the sets of coefficients using rank correlation procedures described in [7]. The remaining 4 steps are applied to each realization.
4) The residuals to the regression surface are transformed using the "Normal-scores transform" as recommended in [8].
5) The transformed residuals can then be used to estimate parameters of the spatial covariance function given above. We used a maximum likelihood procedure to

estimate the parameters for the combined set of residual (spatial difference) data from all realizations. Differences in residual magnitude and distribution are maintained through the transformation and back-transformation in 4) and 6).

6) The "sequential-Gaussian" conditional simulation procedure is used to generate the random function component $\varepsilon(\mathbf{x})$. More detail (of a mechanical nature) on the sequential Gaussian approach is given in [8] Chapter V. The algorithm generates a response surface over the grid in transformed space and then back-transforms the values according to a set of tables constructed during the transformation in 4). The conditioning data are the transformed residuals to the polynomial surface

7) The back-transformed random function term is added to the polynomial surface to complete the realization.

We illustrate some of these steps using the analytical example. Figure 3 shows the construction for two realizations. The regression component is in Figure 3a, the random function component in 3b and the completed realization in 3c. The pair of realizations illustrate possible differences within the ensemble. Figure 4 provides further illustration with three additional completed realizations for this example.

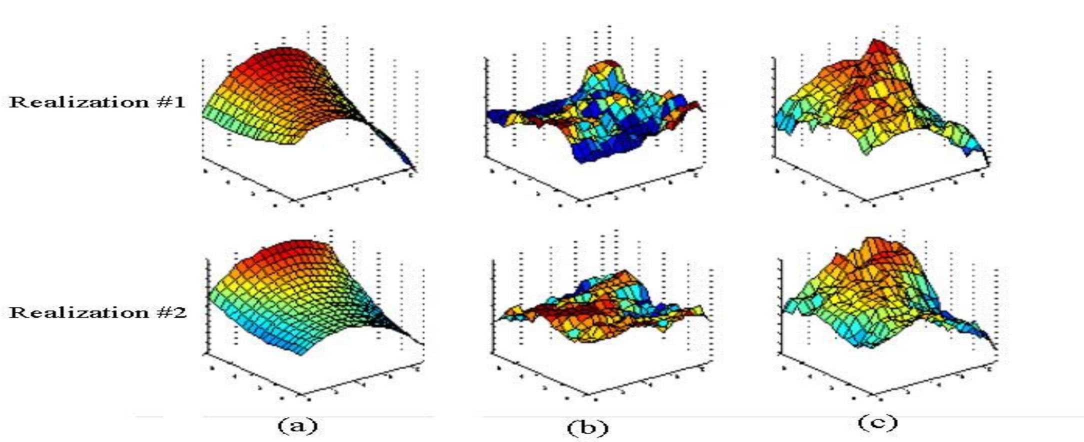

Figure 3. Components of the response realizations for the analytical example.

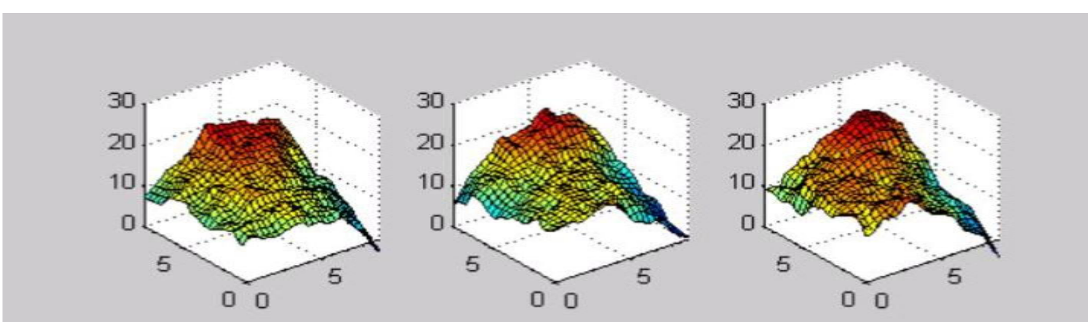

Figure 4. Three of twenty-five realizations constructed for the analytical example.

2.3. Response Model Application

Once the response model has been constructed, it can be used as described earlier to evaluate arbitrary events based on the response. The events of primary interest here relate to system performance. We demonstrate how probabilities for three different performance measures can be assessed using the response model generated for the analytical example. The procedure for the first and last performance measure is to apply the formulas given early using the equally weighted realizations for $G(\mathbf{r})$. We illustrate the performance measure distribution here by showing the corresponding histograms.

Consider, first, the “best-case” performance measure where we will assume high values of the response are “good”. Figure 5a shows a histogram of realization results for this quantity – from this we can easily approximate its distribution. In Figure 5b, the locations in the input space where these maximum values occurred are plotted. This plot addresses the optimization problem where we are interested in the input location yielding the maximum value. The next step might be to quantify in some way (some measure based on clustering metrics, for example) the spread of probable input locations. The maximum values in this example are confined to the discrete grid used to record values for $G(\mathbf{r})$. The final performance measure considered here is reliability where we assume the failure region R^* is that part of the response space exceeding 18. We need distributions concerning the inputs a and b to determine a probability of failure. For simplicity, we arbitrarily choose to assume both input parameters are uniformly distributed over their ranges. Figure 5c provides a histogram for reliability under these assumptions.

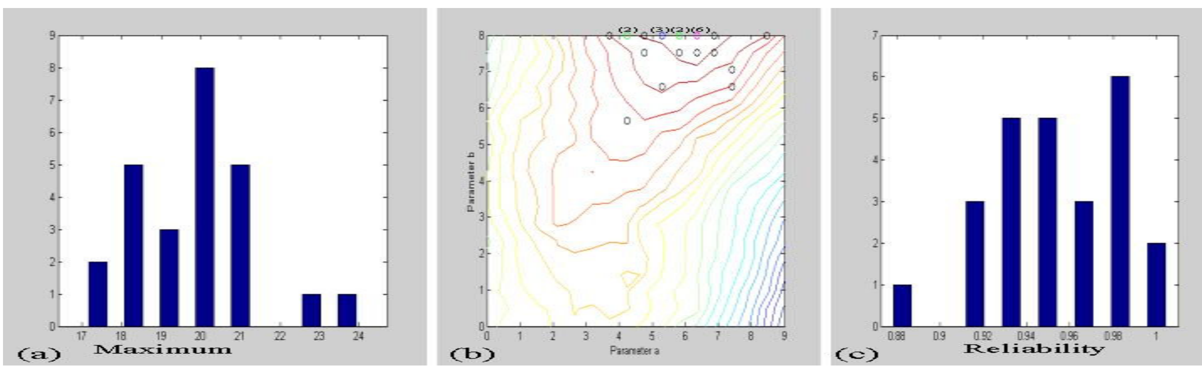

Figure 5. Performance measures for the analytical example.

3. EXTENSIONS TO FUNCTIONAL COMPUTER RESPONSES

Computer responses may be scalars or vectors but can also be functions of time and/or space. The examples that follow illustrate how these more complicated responses can be addressed using the response modeling approach. We refer to the relatively complicated empirical models used to create the probability measures for these responses as “behavioral models.” They combine multiple components, including the response models and a “response assembly” model. For the examples considered here, the response assembly model creates a pulse over time based on a discrete set of “intermediate” parameter values that are modeled using one response model each. The Device #1 example illustrates a case where the response can be modeled adequately through a four-parameter circuit model. In this example the response models are used to approximate probability measures for these four parameters. The Device #2 example gives a case where there is no circuit model that showed the flexibility needed to accommodate the range of pulses in the data set. In this case, basis functions were established using principle components analysis and the response models were used to approximate probability measures for the basis function coefficients. Before proceeding to the examples we outline briefly the process of constructing the behavioral models.

1. Determine a functional form that has the flexibility to accommodate the range of responses anticipated in the application (where possible). The parameters needed to fully specify a response are referred to as intermediate parameters (assume q of them).
2. Determine the "best fit" values of the intermediate parameters for the computer generated conditional response data. This setup will result in a set of p dimensional input -- q dimensional response (intermediate parameter) data.
3. Use the data above to construct q response models. The response models should be p-dimensional unless an analysis of the data indicates that some of the inputs are not important for some of the responses.

The behavioral models can now be used to make predictions. Any specific set of inputs will yield k intermediate parameter values from each of the q response models. These k^q sets of intermediate parameters can be used to generate a distribution of the response corresponding to the specified inputs. If the performance measure is constructed from the sets of responses, k^q values are available to approximate a distribution.

3.1. Device #1 Example

This example provides an illustration of how response modeling can be used for more complex responses and performance criteria. Figure 6 illustrates the behavioral model used in the analysis. The output pulses were modeled using a capacitance discharge unit (CDU) mapping voltage (V_0), inductance (I), capacitance (C) and resistance (R) to the current pulse. Specification of the performance measure, for this example depends on those aspects of the pulse considered critical for performance. Maximum current, for example, is one possible quantity of interest.

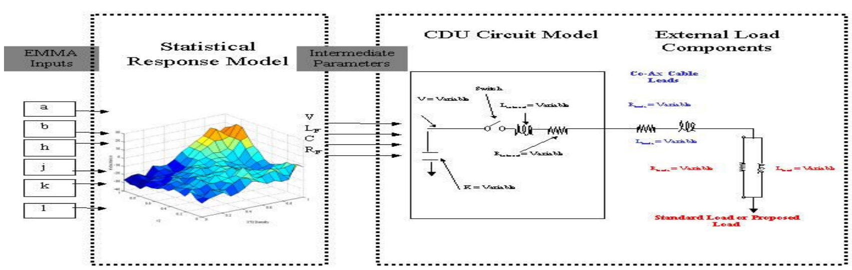

Figure 6. Behavioral model for the Device #1 example.

Following the outline specified above, four response models were constructed for the intermediate parameters (the four electrical parameters). Two objectives of this analysis were to be able to make predictions for arbitrary sets of inputs and to investigate performance throughout the 6-dimensional input space economically. Prediction uncertainty (expressed through 88% bounds) and computer generated values (the circles) for one of the pulses that was not used in constructing the response models are shown in Figure 7a. The confidence bounds apply "point-wise" for individual time values. Six of the twenty-seven points in three pulses used are outside the 88% confidence bounds -- almost twice the number that should fall outside for a typical analysis on average. This result is not unlikely given the high correlation among points and the small sample size of these curves. It is possible, however, that the uncertainty is understated because of assumptions concerning the behavioral model form. These assumptions are difficult to evaluate and this source of uncertainty is not included in computation of the bounds.

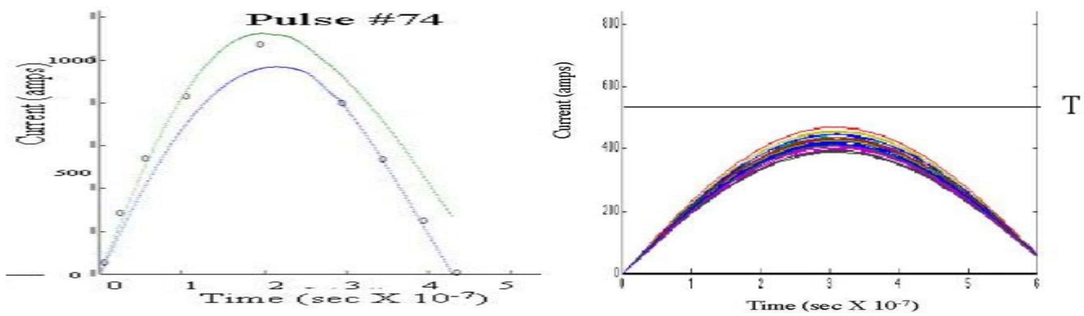

Figure 7. Modeling uncertainty for fixed inputs for the Device #1 example. Figure 7a shows an actual response (the circles) and 88% prediction bounds. Figure 7b shows the pulse yielding the lowest peak current and its related uncertainty.

To illustrate a possible scenario addressing the second objective, consider a hypothetical threshold established at T = 550 amps and assume we are concerned that pulses that do not achieve this threshold may indicate unacceptable reliability. The behavioral model was used to investigate the entire 6-dimensional input space in several hours. Figure 7b shows the ensemble of output pulses (indicating the prediction and uncertainty associated with the prediction) corresponding to the inputs yielding the worst performance according to this criteria.

3.2. Device #2 Example

In the Device #1 example, the current pulses were of shapes that could be accommodated using a circuit model. In this example, no simple formulation, flexible enough to characterize all current pulses, was available. We include this example to illustrate how a set of basis functions can be constructed to provided the flexibility for modeling arbitrary curves or surfaces using a response-modeling approach.

The initial data consisted of 136 runs of the computer model. The current pulses for these runs are shown in Figure 8a. The pulses were "discretized" (to 31 points along the time axis) and were shifted, "time registered", in a way that minimized their squared differences compared with the average pulse (see Figure 8b). The average pulse was then subtracted leaving the residual curves in Figure 8c.

The resulting sets of discrete values were evaluated through principle component analysis see [9] Chapters 5 and 6 for a complete description of these methods. Basis functions, using the principle components, were constructed as described in that text. Figure 9 illustrates the Device #2 behavioral model. Given values for the six inputs, the eight response models each generate 20 values that span the range of coefficients for the appropriate basis function or the time-registration parameter. Performance assessment for this example could be addressed in a similar way to that illustrated for the Device #1 example.

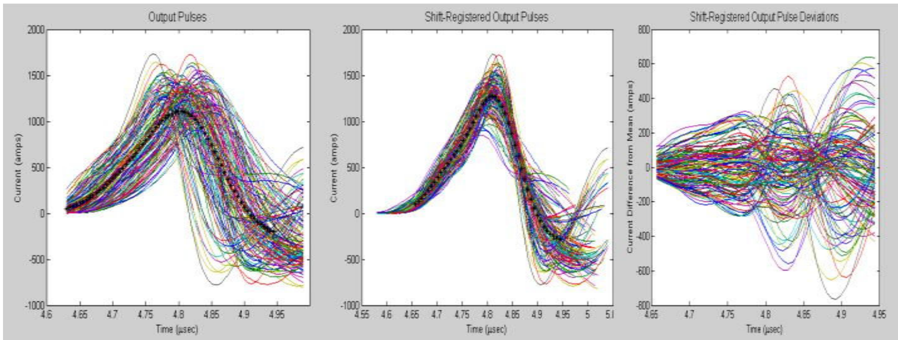

Figure 8. Current pulse responses for the Device #2 example.

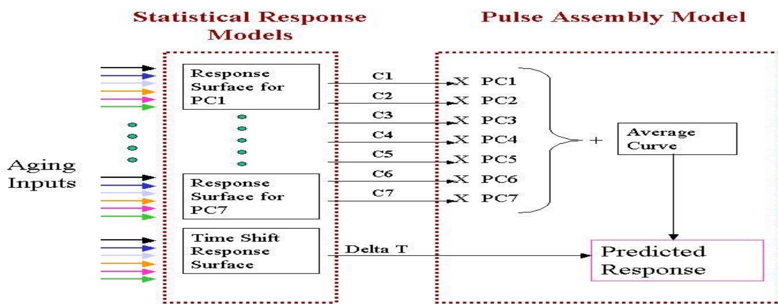

Figure 9. Behavioral model for the Device #2 example.

REFERENCES

1) Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. Design and Analysis of Computer Experiments, *Statistical Science*, Vol. 4, No. 4, 409-423, 1989.

2) O'Connell, M. A., and Wolfinger, R. D. Spatial Regression Models, Response Surfaces, and Process Optimization, *Journal of Computational and Graphical Statistics*, Vol. 6, No. 2, 224-241, 1997.

3) Kennedy, M. C., and O'Hagan, A. Bayesian Calibration of Computer Models (with discussion). *Journal of the Royal Statistical Society (Series B)*, 58, 425-464, 2002.

4) McKay, M. D., Beckman, R. J. and Conover, W. J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, Vol. 21, No. 2, 239-245 1979.

5) Rutherford, B. *An Algorithmic Approach to Experimental Design for the Computer Analysis of a Complex System*, SAND2000-0871, Sandia National Laboratories, Albuquerque, New Mexico, 2000.

6) Rutherford, B. *A Response-Modeling Approach to Uncertainty Characterization and Computer Experimental Design*, Whitepaper, Sandia National Laboratories, Albuquerque, New Mexico, 2003.

7) Iman, R. L. and Conover, W. J. A Distribution-Free Approach to Introducing Rank Correlation Among Input Variables, *Communications in Statistics, Part B - Simulation & Computation*, 11(3), 311-334 1982.

8) Deutsch, C. V. and Journel, A. G. *GSLIB Geostatistical Software Library and Users Guide Second Edition*, Oxford University Press, New York, 1998.

9) Ramsay, J., O. and Silverman, B., W. *Functional Data Analysis* Springer-Verlag, New York, 1997.