Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

ISSUES IN DEVELOPMENT OF SIMULTANEOUS FORWARD-INVERSE METAMODELS

Russell R. Barton

The Smeal College of Business
406 Business Building
The Pennsylvania State University
University Park, PA 16803, U.S.A.

ABSTRACT

Metamodels provide estimates of simulation outputs as a function of design parameters. Often in the design of a system or product, one has performance targets in mind, and would like to identify system design parameters that would yield the target performance vector. Typically this is handled iteratively through an optimization search procedure. As an alternative, one could map system performance requirements to design parameters via an inverse metamodel. Inverse metamodels can be fitted 'for free,' given an experiment design to fit several forward models for multiple performance measures. This paper discusses this strategy, and some of the issues that must be resolved to make the approach practical.

1 MOTIVATION

The design of manufacturing, service and business processes is aided by the availability of discrete-event simulation models of these processes. The models permit rapid inexpensive evaluation of alternative process or system designs, permitting artificial experimentation to a degree well beyond what might be practical for real systems.

Although less expensive to evaluate than the real systems they represent, simulation models can sometimes require extensive computational time. Using design of experiments methods, simulationists can build metamodels: fast-running approximations to the input-output relations exhibited by the original simulation (Kleijnen 1975).

If we represent the output of the simulation model as the random vector Y, then the key output characteristics are generally statistical functions of Y, often the expected value. In this common case, the input-output relations of interest are represented by the vector-valued function $\boldsymbol{f}$:

$$\boldsymbol{f}(x) = \mathrm{E}(Y), \qquad (1)$$

where x is the k-dimensional deterministic vector of design parameters and Y is the p-dimensional random vector of simulation outputs. Using an N-row matrix $\boldsymbol{X}$, row i of which is a vector of design parameter values used in the i^{th} original model run, and a matrix $\boldsymbol{Y}$, each row of which corresponds to a run and each column to a particular component of the output performance vector, a (vector-valued) approximation model $\boldsymbol{m}_f$ is fitted. The objective is to have $\boldsymbol{m}_f(x) \approx \boldsymbol{f}(x)$ for any x in the prediction region R_x. The runs used to fit $\boldsymbol{m}_f(x)$ are restricted to a space C_x. Often $R_x = C_x$.

This metamodeling strategy has many benefits, including improved ability of designers to find good designs interactively (Barron et al. 2004). Unfortunately, the direction of this map (that is, from x-space to y-space) is opposite that of decision-making for customer-driven design. This approach was developed by the Japanese, and described in Hauser and Clausing (1988) in their paper on Quality Function Deployment (QFD). Figure 1 shows a simplified representation of the second house in the four-house representation of QFD. The objective at this stage is to map customer-driven technical specifications for the process into particular values for process design parameters.

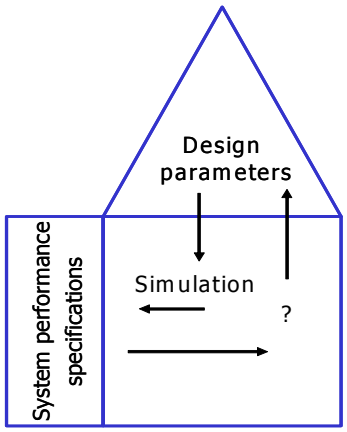

Figure 1: Mapping Directions for the Second House in QFD

This customer-driven view can also be seen as a performance-driven view, and is at the core of Nam Suh's *Axiomatic Design* (Suh 1998). For the great majority of engineering simulation models (discrete-event, finite-element, circuit analysis, etc.), the mapping is from design parameters to performance, rather than vice versa. That is, we have maps $y = \boldsymbol{f}(x)$ but the customer-driven design paradigm requires the map $x = \boldsymbol{f}^{-1}(y_{desired})$. One would ideally map from customer needs through to design parameters through to the implications for manufacturing and delivery that would result (Aungst, Barton, and Wilson 2003).

Product and process design typically involve multiple performance objectives, rather than just one. For example, cost, quality and responsiveness (speed of service) are three general classes of performance measures affecting the design of a service process, and each of these may have several subcategories. There may be additional performance measures (e.g., hours per server) that are used in design constraints (e.g. work rules). On the one hand, these multiple performance measures make design complicated. On the other, the multiple components of Y in Equation (1) are what provide the hope of an inverse map.

Metamodels play a key role in the development of inverse mappings, to further enrich the design assistance that can be provided by simulation. Under certain conditions, the same set of run matrices ($\boldsymbol{X}$, $\boldsymbol{Y}$) used to estimate $\boldsymbol{m}_f$ can be used as ($\boldsymbol{Y}$, $\boldsymbol{X}$) to fit $\boldsymbol{m}_{f^{-1}}$. Suh (1998) imposed the conditions that i) a set of target customer attributes (pre-y's) can be mapped to certain performance targets (for y's), which in turn can be mapped to design parameter values (x's), and ii) that the corresponding (metamodel) mapping functions are linear. One can extend Suh's strategy and consider higher order polynomials as mapping functions, or more general classes of metamodels (Barton 1992), and try to understand process design situations where the design targets may not be obvious.

There are a number of issues that must be resolved in order to take advantage of metamodel-based inverse maps.

1. When the target point $y_{desired}$ occurs at the local minimum or maximum of one or more elements of $\boldsymbol{f}$, the function may not be (locally) invertible.
2. When the dimension of $\boldsymbol{y}$ does not match the dimension of $\boldsymbol{x}$, how can an inverse function be established?
3. How can one find an 'optimal' experiment design for simultaneously fitting $\boldsymbol{m}_f$ and $\boldsymbol{m}_{f^{-1}}$?
4. What is the relationship between $(\boldsymbol{m}_f)^{-1}$ and $\boldsymbol{m}_{f^{-1}}$?
5. How can constraints on $\boldsymbol{x}$ and $\boldsymbol{y}$ be included in the experiment design methodology?

These issues are discussed in more detail in the following sections. Section 2 presents a network design example that is used throughout the discussions. Section 3 discusses the issue of invertibility, and strategies to develop invertible maps. The section also gives an example of how invertible maps can be developed when the dimensions of the design and performances spaces don't match. Section 4 discusses previous work in optimal design, describes how this work can be applied to simultaneous design for forward and inverse metamodels, and describes prior work in combined forward-inverse designs. The section includes brief discussions of issues 4. and 5. as well. Finally, the last section summarizes the importance of this area and the research opportunities that it presents.

2 NETWORK DESIGN EXAMPLE

Consider a simple network design situation. Imagine a communication system in which one must choose routing percentages to a particular destination. Suppose that there are three routes (networks) that might be used. One must choose P_1 (the percent to network 1) and P_2 (percent of the remaining information packets that go to network 2) to minimize costs. Costs are composed of \$.005/time unit each packet is in the system, plus a per-packet processing cost c_i that varies by network: \$.03 for network 1, \$.01 for network 2 and \$.005 for network 3. An Arena model for this system is shown in Figure 2. Suppose that 1000 information packets must be processed, and that packet interarrival times have an exponential distribution with mean $= 1/\lambda = 1$ time unit. Suppose that network transit times have triangular distributions with mean E(S) and limits +/- .5 with E(S) = 1, 2, and 3 for networks 1, 2, and 3 respectively.

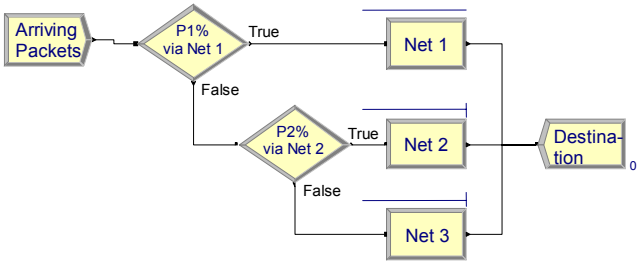

Figure 2: Arena Model of Network Design Example

In terms of the general notation of this paper, $x = (P_1, P_2)'$, and $\boldsymbol{f}$ will have components related to expected delay costs and network use costs. We wish to explore the implications that particular delay and network use costs have on choices of routing probabilities.

3 INVERTIBILITY OF THE SIMULATION MAPPING

To be locally invertible, the function $\boldsymbol{f}$ must be 1-1. Smooth maps will be invertible locally if and only if the matrix of first derivatives, the Jacobian matrix, $J = [\partial f_i/\partial x_j]$

evaluated at that point is invertible (full rank, i.e. have nonzero Jacobian determinant). For the mapping to be globally invertible, the Jacobian determinant must be nonzero everywhere (Chichilnisky 1998). Checking this condition globally is not possible, but a local minimum or maximum for a coordinate function of $\boldsymbol{f}$ implies that all partial derivatives of that coordinate function at that point are zero, and so the Jacobian matrix has a row of zeroes and a zero determinant.

One practical strategy to check for invertibility might be to use a preliminary fitted (forward) metamodel $\boldsymbol{m}$, and check the Jacobian determinant of $\boldsymbol{m}$ either randomly, over a grid, or via a minimization search method to check for locations with Jacobian values of small magnitude.

A practical implication is the need to select the coordinate functions of $\boldsymbol{f}$ carefully, to avoid where possible local minima or maxima, for example. Each f_i should be a monotonic function of the x variables on which it depends. This argues for the decomposition of a total cost function, for example, into separate investment cost and delay cost elements. The separate pieces are monotonically increasing and decreasing, respectively, and both functions of investment (an x), but the sum likely has a minimum at the optimal level of investment (all other design variables held fixed) and so would not be monotonic.

This is illustrated for the network design example in Figures 3 and 4. Figure 3 shows the image of a rectangular grid of x_1 x_2 values in y_1, y_2 space, where y_1 = total delay cost, y_2 = total network use cost and $(x_1, x_2) = (P_1, P_2)$. In this case, delay cost is a sum of delay costs on all three networks. Since P_2 adjusts the relative fraction of traffic between network 2 and network 3, there will be an optimal balance in terms of total delay, resulting in a minimum for y_2 as a function of P_2. Figure 3 shows grid lines that double back on each other, indicating that the map is not 1-1.

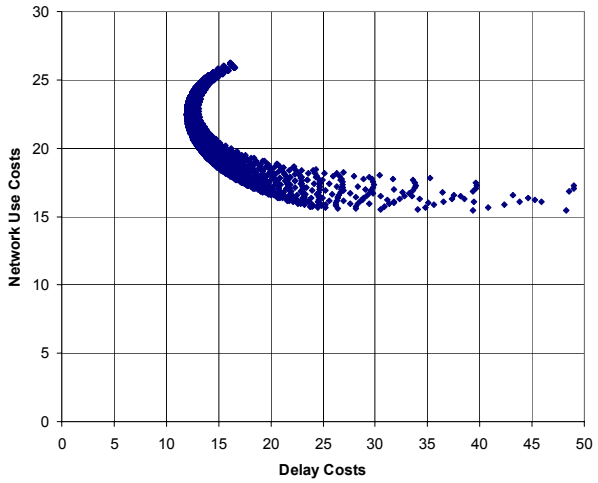

Figure 3: Image of (P_1, P_2) Grid in (*Delay cost, Network Use Cost*) Space

Figure 4 shows a mapping of a (P_1, P_2) grid into total costs for traffic on network 1 and total costs for traffic on network 2. In this case, both the fixed charge per message and the delay cost are increasing functions of the routing percentage, and the plot shows an invertible structure. As for Figure 3, expected costs were estimated using steady state M/G/1 approximations.

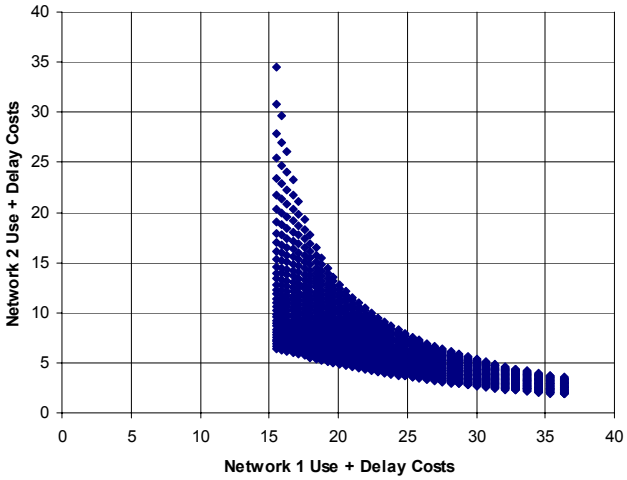

Figure 4: Image of (P_1, P_2) Grid in (*Network 1 Total Cost, Network 2 Total Cost*) Space

This structure involves only two of the three network costs, however. How can one include the costs associated with use of network 3? Let $S = \{(s_1, s_2, s_3)\}$ represent the space of network costs where s_i is the total cost for traffic using network i. Since the x-space has dimension 2 (i.e., (P_1, P_2)), the map from (x_1, x_2) to (s_1, s_2, s_3) generates a surface in S. If the functions are monotonic, any point on this surface can be identified uniquely using the pair (s_1, s_2), with s_3 determined by some function g with $s_3 = g(s_1, s_2)$. One could imagine a user interface that allowed the designer to highlight a point on the (s_1, s_2) plane (below the surface), with the corresponding point on the (s_1, s_2, s_3) surface immediately identified. Using $\boldsymbol{m}_{f^{-1}}$, the corresponding design (P_1, P_2) could be identified simultaneously in an accompanying plot.

4 DESIGN OF EXPERIMENTS ISSUES

To fit $\boldsymbol{m}_f$ and $\boldsymbol{m}_{f^{-1}}$ one must choose a set of simulation runs. There has been a great deal of research on the optimal choice of simulation run conditions to fit metamodels. Some of this work is of general applicability to experimental settings, not just simulation. See for example Silvey (1980), Box and Draper (1987), Atkinson and Donev (1992), Pukelsheim (1993), Myers and Montgomery (1995), and Khuri and Cornell (1996). Other developments have been specific for experimentation with dis-

crete-event dynamic simulation models. For computer simulation experiments see Kleijnen (1987, 1998, 2005), Santner, Williams, and Notz (2003) and the review in Law and Kelton (2000) that describes many other results. Before discussing possible strategies for designing experiments to fit $\boldsymbol{m}_f$ and $\boldsymbol{m}_{f^{\prime}}$ it will be helpful to review the well-known experiment design strategies for multiple regression and other types of metamodels.

4.1 Optimal Design for (Forward) Regression Models

The development of optimal design strategies is most extensive for the standard multiple regression case. The standard multiple regression model captures the following underlying relation:

$$f(x) = \Sigma\beta_q\phi_q(x) + \varepsilon,\ \varepsilon \sim \text{i.i.d. } N(0, \sigma^2). \qquad (2)$$

The response function is modeled as a linear combination of r functions of the k input variables ($q = 0, ..., r$) plus an intercept, with additive, independent homogeneous Gaussian perturbations. For a first order polynomial model, $r = k$, $\phi_q(x) = x_q$, and $\phi_0(x) = 1$. For general multiple regression, there are no restrictions on the form of the ϕ_q functions. For example, $\phi_q(x) = x_5^2$, $\phi_q(x) = \ln(x_3)$, $\phi_q(x) = 1/x_4$ are candidate functions for multiple regression models. The coefficients β_q and random perturbations represented by ε are unknown and are estimated using least-squares or other methods.

The multiple regression metamodel that is constructed assuming a true response of the form shown in Equation (2) is $m_f(x) = \boldsymbol{\phi}(x)'b$. Note that $\boldsymbol{\phi}$, x and b are vectors, and in this case, $m_f(x)$ is a scalar. For the multiple regression model there is a single response. When there are multiple responses, the fitting process can be extended as discussed below. The b vector is calculated using an existing set of ($\boldsymbol{X}$, $\boldsymbol{y}$) data, where x_{ij} is the value of the j^{th} design parameter ($j = 1, 2, ..., k$) in the ith run of the system ($i = 1, 2, ..., N$). Let $\boldsymbol{x}_i$ denote the vector of values for the ith run. Finally, y_i is the (univariate) value of the response in the ith run of the system. Then the least-squares equations can be written in matrix form as

$$b = (\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{D}'\boldsymbol{y}, \qquad (3)$$

where $\boldsymbol{D}$ is the $N \times r$ matrix whose $(i, q)^{\text{th}}$ entry is the value of $\phi_q(\boldsymbol{x}_i)$. The matrix $\boldsymbol{D}$ is called the *design matrix* which is often represented by the letter $\boldsymbol{X}$ in the design of experiments literature. We avoid this notation (and avoid the use of the index j for its columns) due to the obvious confusion with the matrix of design parameter values used in the fitting runs. Even for a first-order (linear) polynomial regression, $\boldsymbol{D}$ and $\boldsymbol{X}$ are not the same; $\boldsymbol{D}$ is augmented with an initial column of ones for the intercept term.

Of course, for many simulation situations, the assumption $\varepsilon \sim$ i.i.d. $N(0, \sigma^2)$ does not hold. In many cases this is because the variance increases with the mean. In some cases it is by deliberate intent, through the use of common and antithetic random numbers, for example. In this case one has $\varepsilon \sim N(\Sigma_Y, \sigma^2)$, where Σ_Y is the variance-covariance matrix for the ε values. The vector β can then be estimated using weighted least squares with $W = (\Sigma_Y)^{-1}$:

$$b = (\boldsymbol{D}'\boldsymbol{W}\boldsymbol{D})^{-1}\boldsymbol{D}'\boldsymbol{W}\boldsymbol{y}. \qquad (4)$$

Alternatively, it is sometimes possible to identify a transformation of the response that produces approximately i.i.d. error. See for example Kleijnen (1987), Cheng, Kleijnen, and Melas (2000), and Chapter 3 of Montgomery (2001). For the remainder of the discussion in this section, the i.i.d. characterization is assumed, permitting estimation using Equation (3) rather than (4).

The form of the model in Equation (2) implies that b can be characterized as a random variable with $E(b) = \beta$. Since b is a vector, it has a variance-covariance matrix, given by

$$\Sigma_{\boldsymbol{b}} = \sigma^2(\boldsymbol{D}'\boldsymbol{D})^{-1} \qquad (5)$$

and with variance of a predicted value at x_0 of b:

$$\text{Var}(m_f(x_0)) = \sigma^2\boldsymbol{\phi}(x_0)'(\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{\phi}(x_0), \qquad (6)$$

with the reminder that the discussion still focuses on a regression metamodel for a single response and so m is not bolded.

In this setting, many characterizations of experiment design goodness minimize some measure associated with (5) or (6). For example, a confidence ellipsoid for the true vector β has a form based on (5), $(\beta - b)'(\boldsymbol{D}'\boldsymbol{D})(\beta - b) = K_\alpha$, where the constant K_α depends on the confidence level desired, $100(1-\alpha)\%$. Minimizing the volume of this ellipsoid corresponds to maximizing the determinant of $(\boldsymbol{D}'\boldsymbol{D})$, which motivates *D-optimality*:

***D-optimality*:** (choose $\boldsymbol{X}$ to) maximize $\det(\boldsymbol{D}'\boldsymbol{D})$, or, equivalently, minimize $\det(\boldsymbol{D}'\boldsymbol{D})^{-1}$.

The optimum design depends on the nature of the functions ϕ_q being fitted and the design points chosen, but surprisingly not on the unknown values of the coefficients. If the optimal design problem is defined as choosing the optimal continuous weighting function on C_x, the optimum can always be achieved using a finite number of distinct x values, at most $r(r + 1)/2$, and often as few as r. This is called the *continuous design problem* (Atkinson and Donev 1992 p. 93). The weights for the points need not be integer or rational so an integer numbers of runs at each point may not be able to match the optimal weights. If the number of

runs at each point must be an integer, the problem is an *exact design problem.* In this case, *D-optimal* designs are solutions to (generally) difficult integer programming problems. These are usually solved approximately using the heuristic methods described in Section 4.5. Similar strategies are employed for other design optimality definitions, including the following.

***G-optimality*:** minimize (max $\{\boldsymbol{\phi}(x_0)'(\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{\phi}(x_0)\}$), where the maximum is over $x \in R_x$.

Again, R_x and C_x are not necessarily the same. The minimization requires x to be in C_x, but the maximization is over x in R_x. N is fixed. For the continuous design formulation, *D-* and *G-optimal* criteria give the same designs, but for the exact design formulation they can differ. A third form minimizes the average variance of the components of b.

***A-optimality*:** minimize $(\Sigma(1/\lambda_q))$, where λ_q, $q = 1, ..., r$ are the eigenvalues of $(\boldsymbol{D}'\boldsymbol{D})$.

A fourth form minimizes the variance of any combination $a'b$ with $\Sigma a_q = 0$ and $a'a = 1$.

***E-optimality*:** minimize (max $\{1/\lambda_q\}$) where the maximum is over $q = 1, ..., r$.

The previous development was for univariate responses. Since $\boldsymbol{f}(x)$ and $\boldsymbol{m}_f(x)$ are vector-valued functions, β and its estimate b are matrices, with one column for each performance measure. We change the notation, replacing b with $\boldsymbol{B}$, and $\boldsymbol{y}$ with the matrix, say $\boldsymbol{Y}$. The least-squares equation, (3), requires little modification:

$$\boldsymbol{B} = (\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{D}'\boldsymbol{Y}. \qquad (7)$$

When the same types of functions (e.g., second-order polynomials in the design variables) are used to approximate each component of the performance vector, then the matrix $\boldsymbol{D}$ has the same form as in the description accompanying Equation (3). Statistical dependencies among components of $\boldsymbol{Y}$ complicate the development of optimal designs. For a discussion of model fitting strategies in this case, see del Castillo, Montgomery, and McCarville (1996). Assuming statistical independence of the ε terms for different $\boldsymbol{Y}$ components and using the same fitting functions for each component of $\boldsymbol{Y}$ puts *D-optimality* in the same form as for the single performance measure case.

Even with a single response, various definitions of design optimality can lead the experimenter to face a multi-objective decision. With multiple responses this is an issue of greater significance. Wong (1999) reviews multi-objective methods for optimal design. Approaches include i) creating an overall objective that is a weighted sum of the individual measures, ii) developing a utility function of more complex form, iii) creating a related 'desirability' function (del Castillo, Montgomery, and McCarville 1996; Kim and Lin 2000), or iv) framing one measure as the objective and the others as constraints. Multiple objectives are also an issue for robust design. Murphy, Tsui, and Allen (2005) discuss a variety of approaches, including utility functions, desirability functions, and as a compromise decision support problem, a specific mathematical programming representation.

4.2 Other Kinds of Model-Based Design Optimality

Since all of the optimality measures described above use $\boldsymbol{D}$ in their formulation, we refer to them as *model-based* measures of design optimality. Because they are based on $\boldsymbol{D}$, their direct application is limited to fitting standard multiple regression metamodels. There are other approaches to optimal design that address the quality of the metamodel approximation but do not focus solely on the information matrix $\boldsymbol{D}'\boldsymbol{D}$ or its inverse, and so can be used with other metamodel types. For example, the *Integrated Mean Squared Error* (*IMSE*) criterion is intended to include both errors of variability and errors of bias caused by assuming the incorrect model form in Equation (2). To create *IMSE-optimal* designs, one must propose an alternative "true" response model, e.g. fitting a first-order model when the true response is quadratic. In fact, one must estimate the value of the unknown coefficients of the extra (e.g., quadratic) terms. Mean squared prediction error criteria also have been used for experiment designs to fit spatial correlation models (Sacks et al. 1989; Currin et al. 1991; Kleijnen and van Beers 2004; van Beers and Kleijnen 2005).

4.3 Model-Free Measures of Design Optimality

Optimal designs based on mean squared error can require extensive numerical calculations. Further, they depend on knowledge of the form of the true response function $\boldsymbol{f}$.

When the model form is uncertain, other measures of design optimality have been developed, in order to generate what are referred to as *model-robust* designs. This situation (as for any model-based experiment design situation) can be approached using decision theory (Berger 1996; Raiffa and Schlaifer 2000). When alternate model forms are postulated, Bayesian priors may be assigned to the form of the correct model. Reviews and algorithms are presented by Heredia-Langner et al. (2004) and Murphy, Tsui, and Allen (2005).

Other model-free design construction strategies focus on the geometric spatial characteristics of the location of the design points in C_x. These include the latin hypercube designs (McKay et al. 1979), orthogonal arrays (Owen 1992), minimax/maximin designs (Johnson, Moore, and Ylvisaker 1990) and uniform designs (Fang and Lin 2003).

4.4 Design Optimality for Forward-Inverse Designs

Design optimality for forward-inverse designs can be based on any of the optimality criteria and approaches described above. For model-based optimality criteria the problem will be multi-objective, since two metamodels are being fitted simultaneously, each with multiple responses. The forward and inverse metamodels will each require a separate fitting operation, because in general, $\boldsymbol{m}_{f^{-1}}(y) \neq (\boldsymbol{m}_f)^{-1}(y)$, even for first-order linear regression metamodels. Equality holds for the special case of orthogonal regression metamodels - see Frank (1971), pp. 38-42. Designs based on spatial optimality may be multi-objective as well, since the optimality measure in x-space may not be directly comparable with the y-space measure.

4.5 Algorithms for Constructing Optimal Forward Designs

Before describing strategies for finding optimal forward-inverse designs, we review construction methods for designs to fit forward-only metamodels. Fundamentally, the alphabetic-optimal designs in section 4.1 can be described as (nonlinear) integer programming problems for the exact design problem and can be formulated as nonlinear programs for the continuous problem. Solution methods generally use heuristics such as exchange methods (Meyer and Nachtsheim 1995) and genetic algorithms (Hamada et al. 2001; Heredia-Langner et al. 2004). One of the first algorithms for exact D-optimal designs was DETMAX (Mitchell 1974), which used a given set of candidate design points and an exchange heuristic to approximate a solution to the IP.

Spatial designs are constructed in a number of ways, but the approaches are generally constructive. That is, an algorithm for generating a set of design points is proved to provide good spatial characteristics. For example, quasi-Monte Carlo methods have been proposed to generate low-discrepancy uniform designs (Fang and Lin 2003). A gridding and random selection procedure generates latin hypercube designs (McKay, Conover, and Beckman 1979). For both of these approaches the design space is expected to be cuboidal or at least (hyper-) rectangular.

These construction algorithms address designs for fitting a single response function, but can be modified to address the fitting of multiple responses (see Murphy, Tsui, and Allen 2005).

All of these algorithms generate a *simultaneous* design, a design that is constructed before any experimental runs are conducted. A potentially more effective (but more difficult) strategy is a *sequential* design, which adds runs sequentially as more is learned about the response function. We will see that this is a critical feature for many forward-inverse design problems.

A number of interesting algorithms for sequential design of experiments were developed to help discriminate between two or more candidate model forms. Hill (1978) gives a review of these early methods. A simple but effective design strategy was proposed by Hunter and Reiner (1965). At the completion of n experimental runs, choose the $(n+1)^{st}$ run at the point x^* that has the greatest difference in predicted value between the two models, i.e. that maximizes $(\boldsymbol{m}_a(x) - \boldsymbol{m}_b(x))^2$.

Santner, Williams, and Notz (2003) developed sequential experiment designs for computer experiments. More recently, Kleijnen and van Beers (2004, 2005) have developed sequential experiment design strategies for discrete-event simulation experiments. Both methods begin with an initial "pilot" experiment design. In their 2004 paper, they used cross-validation on the pilot design (excluding vertices) to provide jackknife estimates of variance at a set of (untested) candidate points, and selected the next run at the candidate point with the highest jackknife variance. In their 2005 paper, the prediction variance is estimated by bootstrap-resampling the simulation outputs, refitting the metamodel, and computing the bootstrap variance of the predicted values at each candidate point. Again, the candidate point with the highest variance is selected as the new candidate.

4.6 Algorithms for Constructing Optimal Forward-Inverse Designs

Optimal design for fitting forward-inverse is complicated by four factors:

1. There are multiple responses.
2. The design spaces in general cannot be represented as rectangular.
3. The design points in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are functionally related and so cannot be chosen independently.
4. It generally will not be possible to evaluate $\boldsymbol{f}^1$ directly.

The first point complicates model-based design strategies, the second complicates spatial design strategies, and the others affect both. The third and fourth factors in particular suggests use of sequential design strategies.

4.6.1 Existing Methods

Barton, Meckesheimer, and Simpson (2000, 2001) proposed one simultaneous and two sequential design methods for finding D-optimal designs for quadratic polynomial forward and inverse models. Method 1, the simultaneous design, allocates half of the N experiment runs in x-space based on the forward D-optimality measure, and half placed in y-space based on the inverse D-optimality measure. The metamodel $\boldsymbol{m}_f$ was fitted using the $N/2$ $\boldsymbol{X}$ results

from the $N/2$ forward design points in x-space and the images in y-space, plus any of the y-space points whose inverse images fell within C_x. The metamodel $\boldsymbol{m}_{f^{-1}}$ was fitted using the $N/2$ D-optimal points in y-space and their corresponding function values (under $\boldsymbol{f}^{-1}$) in x-space, plus any of the x-space design points whose images fell within C_y.

For Method 1, the authors assumed that both forward and inverse models were available, but for the case where the inverse function was not explicitly available, a two-stage sequential strategy (Method 2) was developed based on the same idea. At the first stage of Method 2, the $N/2$ forward and $N/2$ inverse points would be determined based on the D-optimality criterion and the design space constraints C_x and C_y. The $N/2$ forward design points would be used to compute $\boldsymbol{m}_f$, and the same set of ($\boldsymbol{X}$, $\boldsymbol{Y}$) data used to fit a preliminary version of $\boldsymbol{m}_{f^{-1}}$. In Stage 2 this preliminary metamodel was used to find points in x-space corresponding to each of the identified D-optimal points in y-space. The original (forward) simulation code was evaluated these additional $N/2$ inverse image points, resulting in different set of $N/2$ ($\boldsymbol{X}$, $\boldsymbol{Y}$) used to fit the final inverse metamodel $\boldsymbol{m}_{f^{-1}}$.

Note that for the first two methods, the run budget must be such that $N/2$ runs are sufficient for fitting the proposed forward and inverse metamodel types. Method 3 allowed simultaneous fitting of both forward and inverse models with fewer runs by recognizing the design contribution of the forward images to the inverse design. The method used an initial set of N_0 (generally $< N/2$) runs, determined based on D-optimality, to fit preliminary forward and inverse metamodels. The remaining $N - N_0$ runs were chosen in y-space to augment the existing images of the first N_0 runs in a D-optimal way.

As expected, the sequential approaches dominated the first method in terms of prediction performance, and for the simple quadratic example, Method 3 required fewer experiment runs than Methods 1 or 2.

Barton, Meckesheimer, and Simpson (2000, 2001) assumed invertibility for the forward-inverse pair. Lu et al. (1999) developed a forward-inverse fitting method that used a recursive decomposition method to split the design region into invertible subregions, each with the response approximated by a different linear function. Each subregion is characterized by its distribution of performance y-values. Backward mapping is accomplished by first identifying a subregion whose performance distribution is compatible with the given $y_{desired}$, and then using the linear inverse map for that region to determine the x.

4.6.2 New Optimal Design Strategies

The work of Lu et al. (1999) did not identify the experiment design strategy for fitting the metamodels, and the existing work by Barton, Meckesheimer and Simpson only considered D-optimality as the criterion for forward-inverse design. There are many opportunities for other approaches to fitting forward and inverse metamodels. This section discusses some promising approaches that should be investigated.

First, Methods 1, 2, and 3 can be extended to other alphabetic optimality criteria, and via the multicriteria approaches described earlier, to combinations of A-, D-, E-, optimality. Second, Method 3 might be extended to be fully sequential, say Method 4. Rather than use the first-phase inverse metamodel to determine the inverse images of y-space locations of the optimal augmenting runs, the inverse metamodel might be updated after each augmenting run, and the remaining augmenting locations recomputed. The issue of which point to choose first might be handled by a jackknife or bootstrap approach like that of Kleijnen and van Beers (2004) or van Beers and Kleijnen (2005).

More general metamodel forms such as splines and kriging models are typically fitted with spatially optimal designs. Sequential forward-inverse design algorithms can be developed for these kinds of designs as well. For uniform designs, this would require research on an appropriate discrepancy function for non-rectangular design regions (see the y-space regions in Figures 3 and 4). For minimax and maximin designs, the extensions to Model 4 (described above) would be straightforward.

Finally, model-based sequential optimal designs can be used with these more general metamodel forms using sequential design strategies. Building on Model 4, after a first phase fitting design, new points could be added in either forward or inverse space based on bootstrap or cross validation/jackknife estimates of prediction variance. Any new method should address both forward and inverse optimality measures using multicriteria strategies such as those in Murphy, Tsui, and Allen (2005).

5 CONCLUSIONS

Computer simulation models play a key role in the design of products and processes. These models generally provide a map from product/process design variables to product/process performance space. In cases where performance targets are known, it is more convenient to work with the reverse mappings. Metamodeling can provide both forward and inverse approximations from one set of experimental data, but the choice of response variables and the design of the experiment require special considerations. The work in this area has just begun: there are great opportunities to identify effective methods and model types for combined forward-inverse metamodel fitting.

ACKNOWLEDGMENTS

The ideas in this paper draw significantly from the pioneering work of Jack Kleijnen in the areas of metamodeling,

and the design of simulation experiments. The author would also like to thank Martin Meckesheimer and Timothy Simpson for many useful discussions on this topic that resulted in several prior publications. This work was sponsored in part by a research grant from the Smeal College.

REFERENCES

Atkinson, A. C., and A. N. Donev. 1992. *Optimum experimental designs*. New York: Oxford University Press.

Aungst, S., R. R. Barton, and D. T. Wilson. 2003. The virtual integrated design method. *Quality Engineering* 15: 565-579.

Barron, K., T. W. Simpson, L. Rothrock, M. Frecker, R. R. Barton, and C. Ligetti. 2004. Graphical user interfaces for engineering design: impact of response delay and training on user performance. In *Proceedings of DETC'04, the 2004 ASME International Design Engineering Technical Conferences.* American Society of Mechanical Engineers DETC2004/DTM-57085.

Barton, R. R. 1992. Metamodels for simulation input-output relations. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J. J. Swain, D. Goldsman, R. C. Crain and J. R. Wilson, 289-299. Piscataway, N.J.: Institute of Electronic and Electrical Engineers.

Barton, R. R., M. Meckesheimer, and T. W. Simpson. 2000. Experimental design issues for simultaneous fitting of forward and inverse metamodels. In *Proceedings of DETC'00, the 2000 ASME International Design Engineering Technical Conferences.* American Society of Mechanical Engineers DETC2000/DAC-14282.

Barton, R. R., M. Meckesheimer, and T. W. Simpson. 2001. Experimental design issues for simultaneous fitting of forward and inverse metamodels. In *Simulation 2001* (Proceedings of the 4th St. Petersburg Workshop on Simulation), 69-76. St. Petersburg, Russia: NII St. Petersburg University Publishers.

Berger, J. 1996. *Statistical decision theory*. New York: Springer.

Box, G.E.P., and N.R. Draper (1987). *Empirical model-building and response surfaces*. New York: John Wiley & Sons.

Cheng, R. C. H., J. P. C. Kleijnen, and V. B. Melas. 2000. Optimal design of experiments with simulation models of nearly saturated queues. *Journal of Statistical Planning and Inference* 85: 19-26.

Chichilnisky, G. 1998. Topology and invertible maps. *Advances in Applied Mathematics* 21: 113-123.

del Castillo, E., D. C. Montgomery, and D. McCarville. 1996. Modified desirability functions for multiple response optimization. *Journal of Quality Technology* 28: 337-345.

Fang, K.-T., and D. K. J. Lin. 2003. Uniform experimental designs and their Applications in Industry. Chapter 4 in *Handbook of statistics*, vol. 22, ed. R. Khattree and C. R. Rao. 131-170. Amsterdam: Elsevier.

Frank, C. R. Jr. 1971. *Statistics and econometrics.* New York: Holt, Rinehart and Winston.

Hamada, M., H. F. Martz, C. S. Reese, and A. C. Wilson. 2001. Finding near-optimal Bayesian experimental designs via genetic algorithms. *The American Statistician* 55: 175-181.

Hauser, J., and D. Clausing. 1988. The house of quality. *Harvard Business Review* 66: 63-73.

Heredia-Langner, A., D. C. Montgomery, W. M. Carlyle, and C. M. Borror. 2004. Model-robust optimal designs: a genetic algorithm approach. *Journal of Quality Technology* 36: 263-279.

Hill, P. D. 1978. A review of experimental design procedures for regression model discrimination. *Technometrics* 20: 15-21.

Hunter, W. G., and A. M. Reiner. 1965. Designs for discriminating between two rival models. *Technometrics* 7: 307-323.

Johnson, M. E., L. M. Moore, and D. Ylvisaker. 1990. Minimax and maximin designs. *Journal of Statistical Planning and Inference* 26:131-148.

Khuri, A. I., and J. A. Cornell. 1996. *Response surfaces: Designs and analyses*. 2nd ed. New York: Marcel Dekker.

Kim, K.-J., and D. K. J. Lin. 2000. Simultaneous optimization of mechanical properties of steel by maximizing exponential desirability functions. *Applied Statistics* (*JRSS C*) 49: 311-325.

Kleijnen, J. P. C. 1975. A comment on Blanning's metamodel for sensitivity analysis: The regression metamodel in simulation. *Interfaces* 5: 21-23.

Kleijnen, J. P. C. 1987. *Statistical tools for simulation practitioners*. New York: Marcel Dekker.

Kleijnen, J. P. C. 1998. Experimental design for sensitivity analysis, optimization, and validation of simulation models. In *Handbook of simulation*, ed. J. Banks, 173-223. New York: John Wiley & Sons.

Kleijnen, J. P. C. 2005. An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research* 164: 287-300.

Kleijnen, J. P. C., and W. C. M. van Beers. 2004. Application-driven sequential designs for simulation experiments: kriging metamodeling. *Journal of the Operational Research Society* 55: 876-883.

Law, A. W., and W. D. Kelton. 2000. *Simulation modeling and analysis*. 3rd ed. New York: McGraw-Hill.

Lu, S. C.-Y., S. T. S. Bukkapatnam, P. Ge, and N. Wang. 1999. Backward mapping methodology for design synthesis. In *Proceedings of DETC99: 1999 ASME Design Engineering Technical Conference.* American Society of Mechanical Engineers DETC99/DTM-8766.

McKay, M. D., W. J. Conover, and R. J. Beckman. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21: 239-245.

Meckesheimer, M., R. R. Barton, T. W. Simpson, and A. J. Booker. 2002. Computationally inexpensive metamodel assessment strategies. *AIAA Journal* 40: 2053-2060.

Meyer, R. K., and C. J. Nachtsheim. 1995. The coordinate exchange algorithm for constructing exact optimal experimental designs. *Technometrics* 37: 60-69.

Mitchell, T. 1974. An algorithm for the construction of "D-optimal" designs. *Technometrics* 20: 2-3-210.

Montgomery, D. C. 2001. *Design and analysis of experiments,* 5th ed. New York: Wiley.

Murphy, T. E., K.-L. Tsui, and J. K. Allen. 2005. A review of robust design methods for multiple responses. *Research in Engineering Design* 15: 201-215.

Myers, R. H., and D. C. Montgomery. 1995. *Response surface methodology: process and product optimization using designed experiments*. New York: John Wiley & Sons.

Owen, A. 1992. Orthogonal arrays for computer experiments, integration, and visualization. *Statistica Sinica* 2: 439-452.

Pukelsheim, F. 1993. *Optimal design of experiments*. New York: John Wiley and Sons.

Raiffa, H., and R. Schlaifer. 1961. *Applied statistical decision theory*. Boston: Harvard University. Republished, New York: John Wiley and Sons, 2000.

Santner, T. J., B. J. Williams, and W. I. Notz. 2003. *The design and analysis of computer experiments*. New York: Springer-Verlag.

Silvey, S. D. 1980. *Optimal design*. London: Chapman and Hall.

Suh, N. 1998. Axiomatic design theory for systems. *Research in Engineering Design* 10: 189-209.

van Beers, W. C. M., and J. P. C. Kleijnen. 2005. Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. Tilburg University: CentER Discussion Paper No. 2004-63.

Wong, W. K. 1999. Recent advances in multiple-objective design strategies. *Statistica Neerlandica* 53:257-276.

AUTHOR BIOGRAPHY

RUSSELL R. BARTON is a professor in the Department of Supply Chain and Information Systems at Penn State University. He received a B.S. degree in Electrical Engineering from Princeton and M.S. and Ph.D. degrees in Operations Research from Cornell. Before entering academia, he spent twelve years in industry. He is currently serving as president for the INFORMS Simulation Society and program chair for the 2007 Winter Simulation Conference. His research interests include applications of statistical and simulation methods to system design and to product design, manufacturing and delivery. His email address is <rbarton@psu.edu>.