Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

A TWO-PHASE MAXI-MIN ALGORITHM FOR FORWARD-INVERSE EXPERIMENT DESIGN

Russell R. Barton

Dept. of Supply Chain and Information Systems
The Pennsylvania State University
University Park, PA 16802, U.S.A.

ABSTRACT

In customer-driven design of systems or products, one has performance targets in mind and would like to identify system design parameters that yield the target performance vector. Since most simulation models predict performance given design parameter values, this identification must be done iteratively through an optimization search procedure. In some cases it would be preferable to find design parameter values directly via an explicit inverse model. Regression and other forms of approximation 'metamodels' provide estimates of simulation model outputs as a function of design parameters. It is possible to design fitting experiments (DOE's) that allow simultaneous fitting of both forward and inverse metamodels. This paper discusses the potential for this strategy and shows a simple two-phase DOE strategy using a maxi-min measure of DOE quality.

1 INTRODUCTION

Simulation has become an indispensable tool in the design of new products and process, permitting the examination of performance at relatively low cost and risk. Although simulation models are used for design, in most cases they were built for analysis. That is, they predict performance given a set of design parameter values. One might prefer that a design tool would work in reverse: given a set of performance targets, generate a set of design parameter values that provide that performance. Inverse methods have recently become the focus of entire journals in engineering design (Taylor and Francis 2006).

The inverse design approach is also supported by the design approach advocated in *Design for Six Sigma* methodology (Ginn, Streibel and Varner 2004). This Japanese Quality Function Deployment (QFD) methodology was described by Hauser and Clausing (1988) to an American audience, and has become a popular tool for customer-driven design. The role of engineering models in the QFD setting has been described by Ramaswamy and Ulrich (1994) and Aungst, Barton and Wilson (2003). The fundamental concept is that customer-driven design requires an understanding of the propagation of customer needs through technical specifications to design parameters. These design choices propagate through to affect choices for manufacturing processes and management.

Figure 1 shows a simplified representation of the QFD mapping as represented through the four house model. The mapping moves from left-to-right, from a point in customer needs space to a point in performance specification space to a point in product design parameter space to a point in manufacturing design parameter space to a point in manufacturing control space. These five spaces (six in Aungst, Barton and Wilson 2003) are linked by qualitative maps represented by the four houses. In the axiomatic design approach of Suh (1998) the houses are replaced by matrices representing linear transformations, although linear maps are not practical generally.

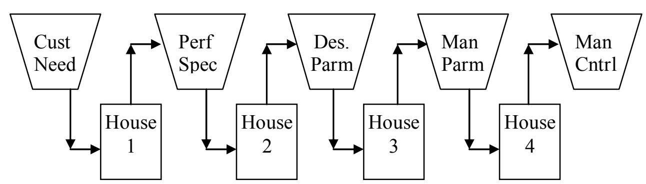

Figure 1: QFD Four House Mapping Representation

The main focus of simulation software is on houses 2 - 4. For example, one can simulate a service process (House 2) or a manufacturing control strategy (House 4). The mapping representations are typically referred to as houses because interactions between variables are represented qualitatively in a triangular 'roof' attached to the square. This structure is illustrated in Figure 2, which provides a more detailed representation. QFD typically represents relationships qualitatively, with the strength of the relationship between a performance specification and a design parameter appearing in the corresponding cell.

This figure also shows that engineering simulation models typically map from design parameters to performance metrics, the opposite direction of customer-driven design. This provides the motivation for this research: to provide quantitative maps from performance specifications to design parameter values. The method builds on the metamodeling strategy employed by simulationists to develop fast-running surrogates for the original simulation models. The term was coined by Kleijnen (1975) and has been a frequent focus of simulation methodology (see Porta Nova and Wilson 1989, Barton 1992, 1998, and Kleijnen and van Beers 2004 for example). Under certain conditions, the experimental data collected to fit forward metamodels can be used in reverse, to fit inverse metamodels that can be used for customer-driven design.

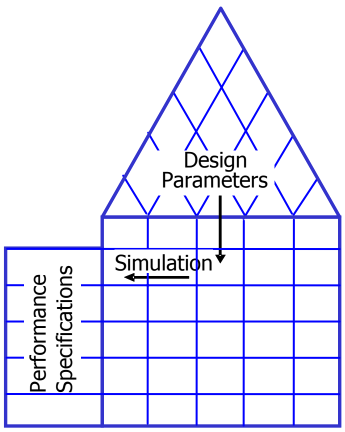

Figure 2: Details for House 2

The next section of this paper describes the metamodeling activity and provides some notation. The inverse issues are illustrated next for a semiconductor manufacturing simulation (Morrice et al. 2005). The overall strategy for building inverse metamodels is described next, along with issues and existing work, and provides a simple two-phase strategy for finding experiment designs that are maxi-min in both domain and range spaces. The next section presents a comparison of the properties of a traditional maxi-min design with a two-phase forward-inverse maxi-min design on two examples. The final section identifies key issues in using forward-inverse metamodels.

2 METAMODELING STRATEGY

If we represent the output of the simulation model as the random vector Y, then the performance measures (to be checked against specifications) 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 DOE 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 $R_x = C_x$.

Metamodels often use the standard multiple regression model. 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 standard multiple regression model there is a single response. When there are multiple responses, the fitting process can be extended by fitting multiple regression models, one for each response. 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 $(i, q)^{th}$ 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 $\varepsilon \sim \mathrm{N}(\Sigma_Y, \sigma^2)$, where Σ_Y is the variance-covariance matrix for the ε values. The vector β can then be estimated using weighted least squares with $W = (\Sigma_Y)^{-1}$:

$$b = (\boldsymbol{D'WD})^{-1}\boldsymbol{D'Wy}. \quad (4)$$

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).

3 DESIGNING INVENTORY POLICY AT FREESCALE

Douglas Morrice and his coauthors described the use of simulation to study job release policy and its impact on inventory and on-time delivery at Freescale Semiconductor, Inc. (Morrice et al. 2005). Figure 3 shows a simplified representation of the operation.

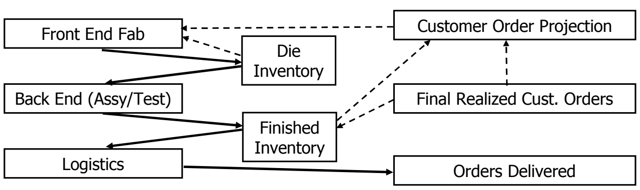

Figure 3: Manufacturing Operations at Freescale

Die inventory levels were used to control job release rates into the front end fab, through the variable MaxDieQ. The front end fab processing time was random, but might be improved by investment in additional equipment to reduce front-end lead time (FELT). Two key performance measures are the (log transform of) fraction of on-time job completions to finished inventory (TFOTD) and the cost associated with inventory and equipment (COST). In our notation, we consider this example to have two design parameters, x_1 = MaxDieQ and x_2 = FELT, and two performance parameters, y_1 = TFOTD and y_2 = COST.

For this example, an inverse model would allow us to explore the cost/on-time-delivery performance space, and choose a Pareto-optimal operating condition. The inverse metamodel would provide the values of die inventory level and reduction in front-end lead time needed to achieve the performance and cost objectives. For the description here, we constructed a quadratic approximation for the TFOTD response based on data in Morrice et al. (2005) and constructed a simple cost function with a cost for lead time reduction of about thirty times the inventory cost savings over the range of design parameter values that were considered.

4 BUILDING INVERSE METAMODELS

Both the simulation models and their approximating metamodels map in the forward direction. That is, we have maps $y = \boldsymbol{f}(x)$ and $y \approx \boldsymbol{m}_f(x)$ but the customer-driven design paradigm requires the map $x = \boldsymbol{f}^{-1}(y_{desired})$. 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 $\boldsymbol{m}_{f^{-1}}$, giving the map $x \approx \boldsymbol{m}_{f^{-1}}(y_{desired})$.

The challenge is to design the set of experimental runs $\boldsymbol{X}$ so that ($\boldsymbol{Y}$, $\boldsymbol{X}$) provides a good set of data for fitting the inverse metamodel. The design used for fitting the forward metamodel does not necessarily place points appropriately for fitting the inverse model. Consider the comparison between Figures 4 and 5. Figure 4 shows a 7 x 7 factorial grid in x-space, and Figure 5 the corresponding image in y-space, based on the Freescale response functions. Design parameters are scaled to +/-1. While the design points are evenly spaced in x-space, they are clumped together in the lower right in y-space, when the most interesting region to explore will be to the right, where there is less focus.

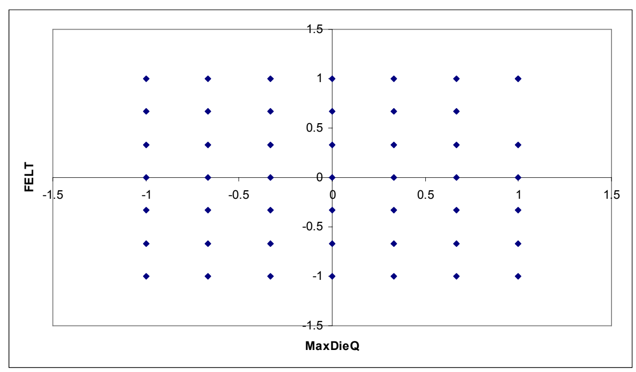

Figure 4: Factorial Grid of Design Points in x-Space

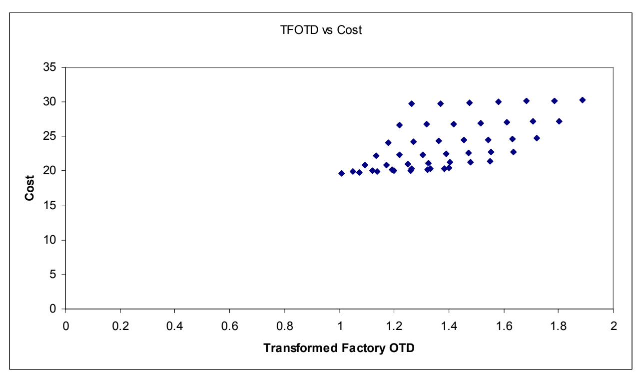

Figure 5: Image of Figure 5 Points in y-Space

The overall strategy for choosing the run conditions will be to seek a set of points that result in good designs both in x-space and y-space. The fundamental steps are:

1. Construct a pilot design in *x*-space (with good properties in that space),
2. Conduct those simulations to fit a pilot metamodel, a pilot inverse metamodel, or both,
3. One or both pilot metamodel(s) then are used to guide the choice of subsequent design points to balance the experiment design quality in both spaces.

Such strategies can be two-phase, with one additional augmenting design in step 3, or fully sequential, with the metamodel(s) updated after each new experiment design point is chosen and run. In the latter case, steps 2 and 3 are iterated with each new experiment design point Multiple-phase strategies between these two extremes are also possible.

In addition to the number of phases, the specific method depends on the metric for design quality. For regression metamodels, alphabetic optimality measures such as those discussed in Silvey (1980) make sense. These depend on the structure of the matrix *D* in *x*-space and its counterpart in *y*-space. Two-phase design methods for D-optimality are described in Barton, Meckesheimer and Simpson (2000, 2001) and Barton (2005).

There are other approaches to optimal design that address the quality of the metamodel approximation, but do not focus solely on the information matrix $\boldsymbol{D'D}$ or its inverse. These can be used with other metamodel types. Some other commonly used measures of design goodness are described in Barton (2005).

This paper focuses on one such measure, the so-called maxi-min experiment designs (Johnson, Moore, and Ylvisaker 1990). The maxi-min criterion maximizes the minimum distance between any two points (sets of run conditions) in the experiment design space. These designs are useful when the metamodel is not a standard regression, and/or when the design region is irregular. Keeping design points far apart is particularly important for spatial correlation models (see Sacks et al. 1989 and Salagame and Barton 1997). The maxi-min strategy for combined forward-inverse designs is proposed below:

1. Use $N_0 \geq N_{min}$ maxi-min (in terms of *x*) first phase forward model runs ($\boldsymbol{X}^{Mmx}$) to generate image points in *y*-space ($\boldsymbol{Y}^{1}$). N_{min} is the minimum number of runs required to fit the chosen metamodel type.
2. Scale $\boldsymbol{X}$ and $\boldsymbol{Y}$ to +/-1 for each coordinate. Keep this scaling through the rest of the process.
3. Fit the phase 1 forward metamodel, $\boldsymbol{m}_{f1}$.
4. In second phase, select $N - N_0$ design points ($\boldsymbol{X}^{Mmxy}$) that are maxi-min both in terms of $\boldsymbol{X}$ (direct calculation) and $\boldsymbol{Y}$ (by computing the distances for candidate image points $\boldsymbol{Y}^2$ using $\boldsymbol{m}_{f1}$).
5. Evaluate the models at $\boldsymbol{X}^{Mmxy}$ to get the true $\boldsymbol{Y}^2$.
6. Fit the final forward metamodel ($\boldsymbol{m}_f$) with $\{\boldsymbol{X}^{Mmx}, \boldsymbol{Y}^1\}$ U $\{\boldsymbol{X}^{Mmxy}, \boldsymbol{Y}^2\}$.
7. Fit the final inverse metamodel ($\boldsymbol{m}_{f\text{-}1}$) with $\{\boldsymbol{Y}^1, \boldsymbol{X}^{Mmx}\}$ U $\{\boldsymbol{Y}^2, \boldsymbol{X}^{Mmxy}\}$, for *y* in $\boldsymbol{Y}^1$ and *y* in $\boldsymbol{Y}^2$ satisfying $y \in C_y$.

The scaling in step 2 is important to allow comparability of the maxi-min design objective in *x*-space and *y*-space. It implies that two points a distance *d* apart in *x*-space are assessed the same figure of merit as two points *d* units apart in *y*-space. Without this scaling, some other method would be required to simultaneously optimize the x-space and y-space designs. Wong (1999) reviews multi-objective methods for optimal experiment 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.

5 MAXI-MIN DESIGN FOR TWO EXAMPLES

The maxi-min strategy described in Section 4 is applied to two examples in this section: the Freescale model described in Section 3, and the network routing example described in Barton (2005). Figure 6 shows the maxi-min design in *x*-space for a 20-point experiment, with 10 points used in the first-phase design. These are coded in blue. Four of the first-phase points were chosen as the extreme points of the design region, and the remaining six chosen as maxi-min in *x*-space. The ten second-phase points maximize the minimum distance from each other and from the ten first-phase points, and are coded in light red.

Figure 7 shows the corresponding *y*-space points, with first-phase points in blue and second-phase points in light red. No points are close together, except for two points in the lower left, which were constructed in the first-phase, using just the *x*-space maxi-min criterion. Compare the quality of this design with the *x*- and *y*-space designs of Figures 8 and 9, respectively, which were constructed using the maxi-min criterion applied only to the *x*-space measures. While the *x*-space design appears slightly better than that in Figure 6, the *y*-space design is significantly worse, with three pairs of closely spaced design points along the bottom of the design region.

Maxi-min design qualities are summarized in Table 1, both for the Freescale example and the network design example in Barton (2005). The two-phase design strategy is labeled "Maxi-Min *x, x+y*." It provides much better maxi-min measures in *y*-space than the standard maxi-min design based solely on *x*-space measures. Further, focusing solely on the *y*-space distances in the second phase does not provide much better maxi-min values in *y*-space for these examples, while the *x*-space performance is noticeably worse.

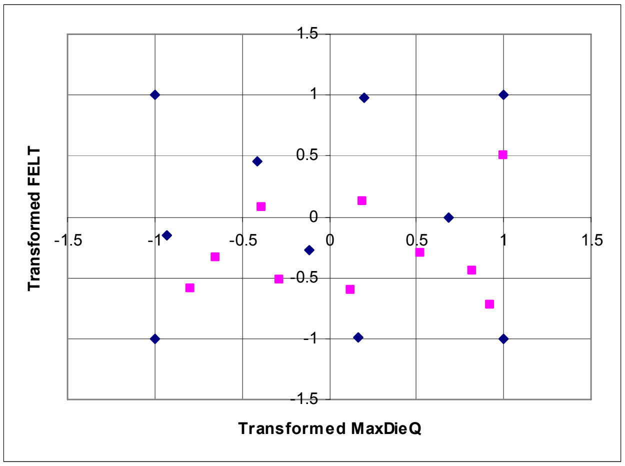

Figure 6: Two-Phase Maxi-Min Design in x-Space

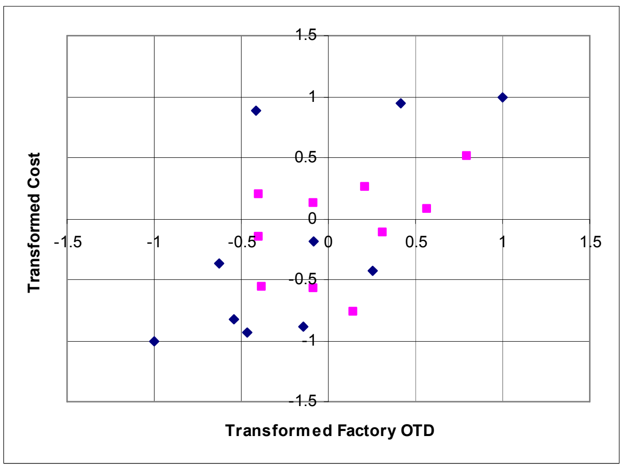

Figure 7: Two-Phase Maxi-Min Design in y-Space

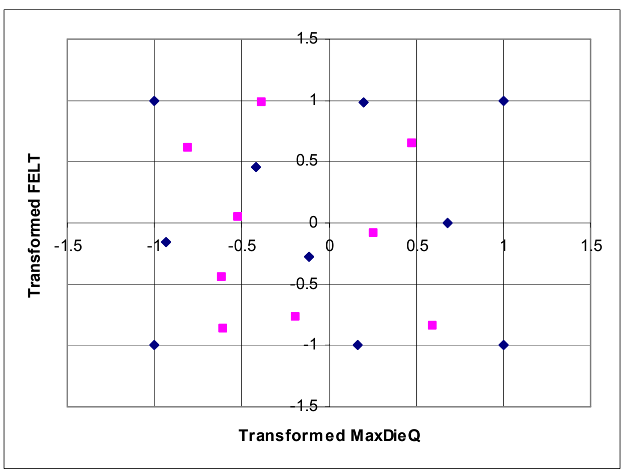

Figure 8: Maxi-Min (on x only) in x-Space

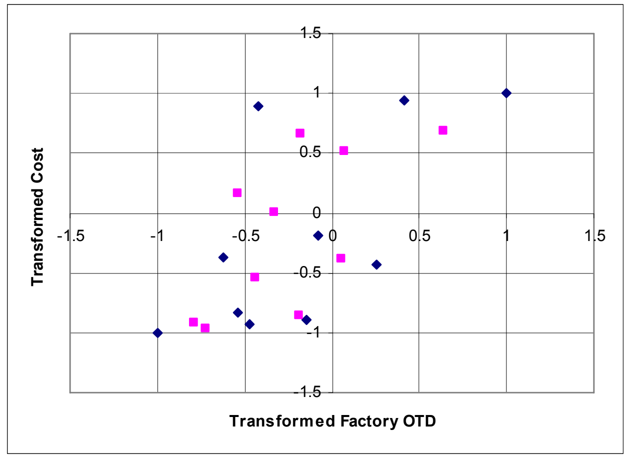

Figure 9: Maxi-Min (on x only) in y-Space

Table 1: Two-Phase Maxi-Min Design Quality

Example	***Design Strategy***	***Min x***	***Min y***
Freescale	Maxi-Min x	0.41	0.05
	Maxi-Min $x, x+y$	0.29	0.29
	Maxi-Min x, y	0.22	0.30
Network Design	Maxi-Min x	0.38	0.08
	Maxi-Min $x, x+y$	0.14	0.14
	Maxi-Min x, y	-	-

6 CONCLUSIONS

Customer-driven design suggests a need for simulation models that move from performance specification to the design parameter values that will generate that performance. Engineering simulation models operate in the wrong direction for this, but the mechanism for constructing metamodels can be applied equally well for fitting inverse maps. It is important to design the fitting experiment with this purpose in mind, however, since the optimal design for fitting the forward model can be far from optimal for fitting an approximation to the inverse map.

The simple two-phase maxi-min strategy described in Section 4 is effective for constructing designs that have good properties in both x- and y-space. The quality of the second-phase points will depend on the quality of the first-phase metamodel, and so it is important to choose an appropriate metamodel type and to make simulation run length long enough to minimize errors in the fitted metamodel.

Many issues remain in developing a generally applicable methodology for constructing inverse metamodels. These issues include defining the design variables, performance measures and experimental region so that the map is invertible, dealing with irregular design regions, and handling statistical issues in fitting (inverse) metamodels with errors-in-variables data. Some of these issues were discussed in more detail in Barton (2005).

ACKNOWLEDGMENTS

The author would 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

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

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. 1998. Simulation metamodels. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, 167-174. Piscataway, N.J.: Institute of Electronic and Electrical Engineers.

Barton, R. R. 2005. Issues in development of simultaneous forward-inverse metamodels. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 209-217. 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.

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.

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

Ginn, D., B. Streibel and E. Varner. 2004. *The Design for Six Sigma Memory Jogger*. Salem, NH: GOAL/QPC.

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

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

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., 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.

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

Morrice, D. J., R. A. Valdez, J. P. Chida, Jr., and M. Eido. 2005. Discrete event simulation in supply chain planning and inventory control at Freescale Semiconductor, Inc. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1718-1724. Piscataway, N.J.: Institute of Electronic and Electrical Engineers.

Porta Nova, A. M., and J. R. Wilson. 1989. Estimation of multiresponse simulation metamodels using control variates. *Management Science* 35: 1316–1333.

Ramaswamy, R. and K. T. Ulrich. 1994. Augmenting the house of quality with engineering models. *Research in Engineering Design* 5: 70-79.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. Design and analysis of computer experiments. *Statistical Science* 4: 409-435.

Salagame, R. and R. R. Barton. 1997. Factorial hypercube designs for spatial correlation regression. *Journal of Applied Statistics* 24: 483-503.

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.

Taylor and Francis. 2006. *Inverse Problems in Science and Engineering*. Listed at http://www.tandf.co.uk /journals/titles/17415977.asp. Accessed 4/10/06.

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 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>.