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

KRIGING METAMODELING IN DISCRETE-EVENT SIMULATION: AN OVERVIEW

Wim C.M. Van Beers

Department of Information Systems and Management
Tilburg University
Postbox 90153, 5000 LE Tilburg, THE NETHERLANDS

ABSTRACT

Many simulation experiments require considerable computer time, so interpolation is needed for sensitivity analysis and optimization. The interpolating functions are 'metamodels' (or 'response surfaces') of the underlying simulation models. For sensitivity analysis and optimization, simulationists use different interpolation techniques (e.g. low-order polynomial regression or neural nets). This paper, however, focuses on Kriging interpolation. In the 1950's, D.G. Krige developed this technique for the mining industry. Currently, Kriging interpolation is frequently applied in Computer Aided Engineering. In discrete-event simulation, however, Kriging has just started. This paper discusses Kriging for sensitivity analysis in simulation, including methods to select an experimental design for Kriging interpolation.

1 INTRODUCTION

A primary goal of simulation is 'what-if' or sensitivity analysis: What happens to the outputs if inputs of the simulation model change? Therefore simulationists run a given simulation program—or computer code—for (say) n different combinations of the k simulation inputs and observe the outputs. (Most simulation models have multiple outputs, but in practice these outputs are analyzed per output type.) To analyze this input/output (I/O) data, classic analysis uses low-order regression metamodels; see Kleijnen (1998). A *metamodel* is an approximation of the I/O transformation implied by the underlying simulation program. (In certain disciplines, metamodels are also called: Response surface, compact model, emulator, etc.) Such a metamodel treats the simulation model as a *black box*; that is, the simulation model's I/O is observed, and the parameters of the metamodel are estimated. This black-box approach has the following advantages and disadvantages.

An *advantage* is that the metamodel can be applied to the output of all types of simulation models, either deterministic or random, either in steady-state or in transient state. A *disadvantage* is that it cannot benefit from the specific structure of the simulation model, so it may take more computer time compared with techniques such as perturbation analysis and score functions.

Metamodeling can also help in optimization and validation of a simulation model. This paper, however, does not discuss these two topics. Further, if the simulation model has hundreds of inputs, then special 'screening' designs are needed, discussed in Campolongo, Kleijnen, and Andres (2000). The examples in this paper, however, limit the number of inputs only to one or two.

Whereas polynomial-regression metamodels have been applied extensively in discrete-event simulation (such as queueing simulation), *Kriging has hardly been applied to random simulation*. However, in deterministic simulation (applied in many engineering disciplines; see for example De Geest et al. 1999), Kriging has been applied frequently, since the pioneering article by Sacks et al. (1989). In such simulation, Kriging is attractive because it can ensure that the metamodel's prediction has exactly the same value as the observed simulation output. In random simulation, however, this Kriging property may not be so desirable, since the observed (average) value is only an estimate of the true, expected simulation output.

Note that several types of *random simulation* may be distinguished:

1. Deterministic simulation with randomly sampled inputs. For example, in investment analysis the cash flow development over time can be computed through a spreadsheet such as Excel. Next, the random values of inputs are sampled—such as the cash flow growth rate—by means of either Monte Carlo or Latin Hypercube Sampling (LHS) through an add-on such as @Risk or Crystal Ball; see Van Groenendaal and Kleijnen (1997).
2. Discrete-event simulation. For example, classic queueing simulation is applied in logistics and telecommunications; see Van Beers and Kleijnen (2003).

3. Combined continuous/discrete-event simulation. For example, simulation of nuclear waste disposal represents the physical and chemical processes through deterministic non-linear difference equations and models the human interventions as discrete events (see Kleijnen and Helton, 1999).

The remainder of this paper is organized as follows. Subsection 2.1 sketches the history of Kriging and its application in geology and in simulation. Subsection 2.2 describes the basics of Kriging and gives the formal Kriging model. Section 3 discusses classic designs for Kriging and mentions criteria for measurement of their performance. Subsection 3.1 treats customized designs for Kriging in deterministic simulation, whereas subsection 3.2 treats customized designs for random simulation. Both subsections demonstrate the performance of the customized designs by two academic simulation models. Section 4 presents conclusions and topics for future research.

2 KRIGING

2.1 History of Kriging

In the 1950s, the South African mining engineer *D.G. Krige* (born in 1919) devised an interpolation method to determine true ore-bodies, based on samples. The basic idea is that these predictions are weighted averages of the observed outputs, where the weights depend on the distances between the input location to be predicted and the input locations already observed. The weights are chosen so as to minimize the prediction variance, i.e., the weights should provide a Best Linear Unbiased Estimator (BLUE) of the output value for a given input. Therefore, Kriging is also called Optimal Interpolation.

The dependence of the interpolation weights on the distances between the inputs was mathematically formalized by the French mathematician Georges Matheron (1930-2000) in his monumental 'Traité de géostatistique appliquée' (1962). He introduced a function, which he called a *variogram*, to describe the variance of the difference between two observations. The variogram is the cornerstone in Kriging. Hence, accurate estimation of the variogram, based on the observed data, is essential. Journel and Huijbregts (1978, pp. 161-195) present various parametric variogram models. The values of its parameters are obtained by either Weighted Least Squares (WLS) or Maximum Likelihood Estimation (MLE); see Cressie (1993).

So Kriging originated in geostatistics to answer concrete questions in the gold mining industry: Drilling for ore—deep under the ground—is expensive, so efficient prediction methods are necessary. Later on, Kriging was successfully introduced into deterministic simulation by Sacks et al. (1989). For example, Kriging is nowadays often applied in CAE. Van Beers and Kleijnen (2003) introduce Kriging interpolation into the area of random simulation.

2.2 Formal Model for Kriging

A *random process* $Z(\bullet)$ can be described by $\{Z(\mathbf{s}) : \mathbf{s} \in D\}$ where D is a fixed subset of R^d and $Z(\mathbf{s})$ is a random function at location $\mathbf{s} \in D$; see Cressie (1993, p. 52).

There are several types of Kriging, but this paper limits to *Ordinary Kriging*, which makes the following two assumptions:

1. The *model assumption* is that the random process consists of a constant μ and an error term $\delta(\mathbf{s})$:

$$Z(\mathbf{s}) = \mu + \delta(\mathbf{s}) \quad \text{with} \quad \mathbf{s} \in D, \mu \in R$$

2. The *predictor assumption* is that the predictor for the point $\mathbf{s}_0$—denoted by $p(Z(\mathbf{s}_0))$—is a weighted linear function of all the observed output data:

$$p(Z(\mathbf{s}_0)) = \sum_{i=1}^{n} \lambda_i Z(\mathbf{s}_i) \quad \text{with} \quad \sum_{i=1}^{n} \lambda_i = 1 . \quad (1)$$

To select the weights λ_i in (1), the *criterion* is minimal mean-squared prediction error (MSE), defined as

$$\sigma_e^2 = E\left(\left(Z(\mathbf{s}_0) - p(Z(\mathbf{s}_0))\right)^2\right). \quad (2)$$

Substituting the *variogram*, defined as

$$2\gamma(\mathbf{h}) = \text{var}[Z(\mathbf{s}+\mathbf{h}) - Z(\mathbf{s})],$$

in (2) gives the optimal weights $\lambda_1, \ldots, \lambda_n$

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\gamma} + \mathbf{1}\frac{1 - \mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}}\right)' \boldsymbol{\Gamma}^{-1}, \quad (3)$$

where $\boldsymbol{\gamma}$ denotes the vector of (co)variances $(\gamma(\mathbf{s}_0 - \mathbf{s}_1), \ldots, \gamma(\mathbf{s}_0 - \mathbf{s}_n))'$, $\boldsymbol{\Gamma}$ denotes the $n \times n$ matrix whose $(i, j)^{\text{th}}$ element is $\gamma(\mathbf{s}_i - \mathbf{s}_j)$, $\mathbf{1} = (1, \ldots, 1)'$ is the vector of ones; also see Cressie (1993, p. 122).

Note that these optimal Kriging weights λ_i depend on the specific point $\mathbf{s}_0$ that is to be predicted, whereas linear-regression metamodels use fixed estimated parameters (say) $\hat{\boldsymbol{\beta}}$ for each $\mathbf{s}_0$ to be predicted.

However, in (3) $\gamma(\mathbf{h})$ is *unknown*. The usual *estimator* is

$$2\hat{\gamma}(\mathbf{h}) = \frac{1}{|N(\mathbf{h})|}\sum_{N(\mathbf{h})}(Z(\mathbf{s}_i - Z(\mathbf{s}_j)^2$$

where $|N(\mathbf{h})|$ denotes the number of distinct pairs in $N(\mathbf{h}) = \{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}\,; i, j = 1, \ldots, n\}$; see Matheron (1962).

3 DESIGNS FOR KRIGING

An experimental *design* is a set of *n* combinations of *k* factor values. These combinations are usually bounded by 'box' constraints: with $a_j, b_j \in R$ and $j = 1, \ldots, k$. The set of all feasible combinations is called the *experimental region* (say) *H*. We suppose that *H* is a *k*-dimensional unit cube, after rescaling the original rectangular area.

Our goal is to find the 'best' design for Kriging predictions within *H*; the Kriging literature proposes several criteria (see Sacks et al. 1989, p. 414). Most of these criteria are based on the predictor's MSE (2). Most progress has been made for the IMSE (see Bates et al. 1996):

$$IMSE = \int_H \mathrm{MSE}\left(\hat{Y}(\mathbf{x})\right)\phi(\mathbf{x})d\mathbf{x} \qquad (4)$$

where MSE follows from minimizing (2), and $\phi(\mathbf{x})$ is a given weight function—usually assumed to be a constant.

To evaluate a design, Sacks et al. (1989, p. 416) compare the predictions with the known output values of a *test set* consisting of (say) *N* inputs. Assuming a constant $\phi(\mathbf{x})$ in (4), the IMSE can then be estimated by the Empirical IMSE (EIMSE):

$$EIMSE = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2 . \qquad (5)$$

Besides this EIMSE, we will also study the *maximum* MSE; that is, we also consider risk-averse users (also see Van Groenigen, 2000). So IMSE—defined in (4)—is replaced by

$$MaxMSE = \max_{\mathbf{x} \in H}\left\{MSE\left(\hat{Y}(\mathbf{x})\right)\right\}$$

and EIMSE in (5) by

$$EMaxIMSE = \max_{i \in \{1, \ldots, m\}}\left\{\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2\right\}. \qquad (6)$$

The most popular design type for Kriging is *Latin Hypercube Sample* (LHS). This type of design was introduced by McKay, Beckman, and Conover (1979) for deterministic simulation models. Those authors did not analyze the I/O data by Kriging (but they did assume I/O functions more complicated than the polynomial models in classic DOE). LHS offers *flexible* design sizes *n* (number of input combinations actually simulated) for any *k* (number of simulation inputs). LHS proceeds as follows; also see the example for *k* = 2 factors in Figure 1.

1. LHS divides each input range into *n* intervals of equal length, numbered from 1 to *n* (so the number of values per input can be much larger than in designs for low-order polynomials).
2. Next, LHS places these integers 1,…, *n* such that each integer appears exactly once in each row and each column of the design matrix.
3. Within each cell of the design matrix, the exact input value may be sampled uniformly. (Alternatively, these values may be placed systematically in the middle of each cell. In risk analysis, this uniform sampling may be replaced by sampling from some other distribution for the input values.)

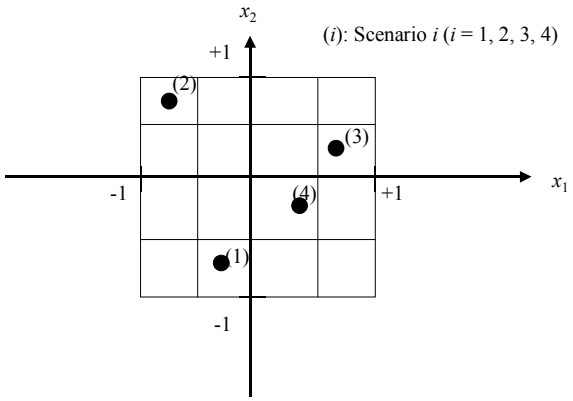

Figure 1: A LHS Design for Two Factors & Four Scenarios

Because LHS implies randomness, its result may happen to be an *outlier*. For example, it might happen—with small probability—that two input factors have a correlation coefficient of –1 (all their values lie on the main diagonal of the design matrix). Therefore the LHS may be adjusted to become (nearly) orthogonal; see Ye (1998).

Classic designs simulate *extreme scenarios*—namely the corners of a *k*-dimensional square—whereas LHS has better *space filling* properties; again see Figure 1. This space filling property has inspired many statisticians to develop related designs. One type maximizes the minimum Euclidean distance between any two points in the *k*-dimensional experimental area. Other designs minimize the maximum distance. See Koehler and Owen (1996), Santner, Williams, and Notz (2003), and also Kleijnen et al. (2004).

3.1 Customized Sequential Designs for Deterministic Simulation

Kleijnen and Van Beers (2004) derive designs that are *customized*; that is, they are not *generic* designs (such as 2^{k-p} designs or LHS). More precisely, these customized designs account for the specific input/output function of the particular simulation function at hand. This customization is achieved through *cross-validation* and *jackknifing*. Furthermore, these designs are *sequential*, because sequential procedures are known to be more 'efficient'; see, for example, Ghosh and Sen (1991) and Park et al. (2002).

The procedure starts with a 'small' *pilot* design of size (say) n_0. To avoid extrapolation, the procedure first selects the 2^k vertices of *H*. Besides these vertices, the procedure selects some extra points—space-filling—to estimate the variogram. After selecting and simulating the pilot design, the procedure selects (say) *c candidate* inputs—again, space-filling—without actually running the simulation model for these candidates. To find the 'winning' candidate, the procedure estimates the variance of the of the predicted output at each candidate input. Therefore, the procedure uses cross-validation and jackknifing. Figure 2 demonstrates the procedure for a fourth-order polynomial simulation model.

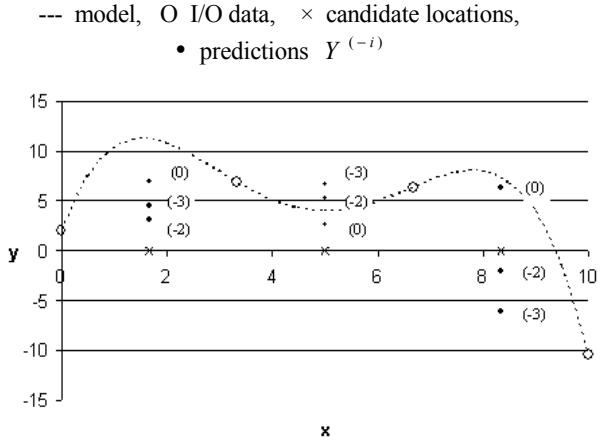

Figure 2: Fourth-Order Polynomial Example, including Four Pilot Observations and Three Candidate Inputs with Predictions Based on Cross-Validation, where (-*i*) Denotes which Observation *i* is Dropped in the Cross Validation

After selecting and simulating the winning candidate, the procedure adds the new observation to the current design. With respect to the augmented design, the procedure selects a new set of candidates. The sequential procedure—selecting a set of candidates, estimating the variance of the predicted output, simulating the winning candidate and augmenting the design—is stopped when a specified criterion is reached. Kleijnen and Van Beers (2004) use the *Successive Relative Improvement* (SRI) after *n* observations:

$$SRI_n = \left| \max_j \{\tilde{s}_j^2\}_n - \max_j \{\tilde{s}_j^2\}_{n-1} \right| \Big/ \max_j \{\tilde{s}_j^2\}_{n-1}$$

where $\max_j \{\tilde{s}_j^2\}_n$ denotes the maximum jackknife variance over $j = 1, \ldots, c$ candidates after *n* evaluations. Note that there are several stopping rules; for example, Sasena et al. (2002) use the *Generalized Expected Improvement* function, which selects inputs that have high model inaccuracy. They stop their tests—rather arbitrary—after 100 calls of this function, whereas Schonlau (1997) proposes to stopping once the ratio of the expected improvement becomes sufficiently small, e.g. 0.01.

Kleijnen and Van Beers (2004) test their Customized Sequential Designs (CSD) through two academic applications:

1. the hyperbolic I/O function $y = \frac{x}{1-x}$ with $0 < x < 1$
2. the fourth-order polynomial I/O function $y = -0.0579x^4 + 1.11x^3 - 6.845x^2 + 14.1071x + 2$ with $0 \le x \le 10$.

To quantify the CSD's performance, they use a test set, consisting of 32 true test values, and compare the Kriging prediction error for the CSD with the prediction error for a LHS design of the same size. Both *EIMSE* and *EMaxIMSE* have substantial smaller values for the CSD than for the LHS designs. Moreover, both examples show that the CSD procedure simulates relatively many input combinations in those sub-areas that have interesting I/O behavior. Figure 3 shows the final design for the fourth-order polynomial example with RSI < 1% and *n* = 24 observations.

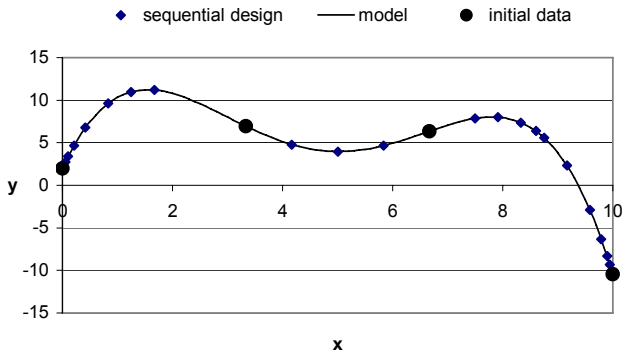

Figure 3: Final Design for Fourth-Order Polynomial Example with *n* = 24 Observations

3.2 Designs for Random Simulation

To select an experimental design for interpolation in random simulation, especially discrete event simulation, Van Beers and Kleijnen (2004) propose a new method. Unlike LHS, the method accounts for the specifics of the model's I/O function. More precisely, the method is customized. To estimate the prediction uncertainty at unobserved input combinations—caused by the noise and the shape of the I/O function—the method uses bootstrapping; i.e., for each scenario already simulated the outputs are re-sampled (for bootstrapping in general see the classic textbook, Efron and Tibshirani 1993; for bootstrapping in the validation of regression metamodels in simulation see Kleijnen and Deflandre, 2005).

Similar to the CSD for deterministic simulation, the procedure starts with a small *pilot* design with input combinations equally spread over the experimental area. Van Beers and Kleijnen (2004) use a *maximin* design, which maximizes the minimum distance between any two points of the design; see Koehler and Owen (1996, p. 288).

Next, for each input value x_i of the pilot design the procedure simulates (say) m_i IID *replicates* until a predefined accuracy level for the estimated output y_i is reached. Then, per input x_i the procedure bootstraps m_i outputs; i.e., the m_i observed outputs per input x_i are re-sampled with replacement. Further, the procedure computes the bootstrap averages $\bar{y}_i^*(m_i)$ per input x_i. (The superscript * indicates a bootstrapped value, as traditional in bootstrap literature.) The re-sampling per input x_i is repeated (say) B times (B is called the bootstrap sample size). Now, the B bootstrapped designs are used to compute B Kriging predictors for the expected outputs of a new set of (say) n^c *candidates*.

To quantify the prediction uncertainty, the procedure computes the *bootstrap variance* for each candidate:

$$\hat{\text{var}}(\hat{y}_j^{c*}) = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{y}_{j;b}^{c*} - \bar{\hat{y}}_j^{c*})^2 \qquad (7)$$

where $\hat{y}_{j;b}^{c*}$ is the predicted value at candidate input x_j^c $(j = 1, \ldots, n^c)$, based on the bootstrapped I/O data $(x_i, \bar{y}_{i;b}^*(m_i))$ $(i = 1, \ldots, n_0)$ and $\bar{\hat{y}}_j^{c*} = \sum_{b=1}^{B}\hat{y}_{j;b}^{c*}/B$. The candidate that has the largest bootstrap variance (7) is added to the current design and simulated until the desired accuracy is reached.

Van Beers and Kleijnen (2004) test the CSD through two classic academic simulation models, namely the *M*/*M*/1 model with one input—the traffic rate (say) ρ—and an (s, S) inventory model with two inputs—the reorder level s and order up-to level S. They compare the performance of the CSD with a LHS design of the same size.

For the M/M/1 example, Figure 4 displays simulation results for both the CSD and the LHS design. The stopping criterion is that $n = 10$ traffic rates have been simulated. The figure shows that LHS simulates fewer 'challenging' inputs; i.e., high traffic rates.

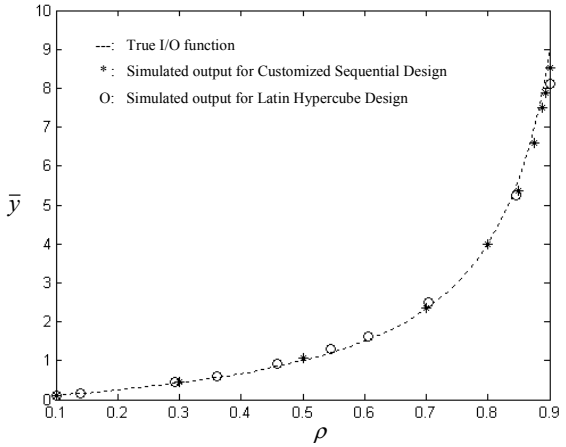

Figure 4: Two Designs for M/M/1 with 10 Traffic Rates ρ and Average Simulation Outputs $\bar{y}$

Van Beers and Kleijnen (2004) use a *test set* with N = 32 equidistant traffic rates (Sacks et al. 1989 also use test sets to evaluate their procedure). They compare the Kriging predictions of the two designs with the 'true' outputs of the test set. Figure 5 illustrates the 32 predictions for the CSD and the LHS design.

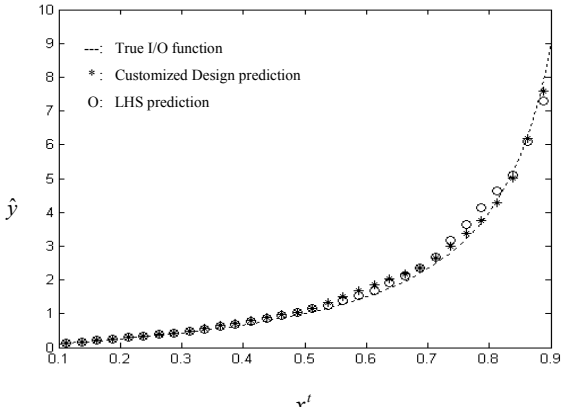

Figure 5: 32 Predictions $\hat{y}$ for the Test Set for M/M/1, for Two Designs

To compare the performance of the CSD design and LHS, they use the *EIMSE* criterion, defined in (5). However, the final numbers of replicates in the two designs may differ, so they calculate the *corrected EIMSE*:

$$CEIMSE = C \times \frac{1}{n^t}\sum_{i=1}^{n^t}\left(\hat{y}(x_i^t) - y(x_i^t)\right)^2, \qquad (8)$$

where C is the ratio of the total number of replicates in the LHS design and in the CSD, n^t is the number of I/O combinations in the test set (so $n^t = 32$), and x_i^t is the i^{th} input of the test set. It turns out that CSD give smaller *CEIMSE* than LHS designs.

To test their procedure for a model with two inputs, Van Beers and Kleijnen (2004) use the (s, S) features of Law and Kelton (2000)'s example 12.9. Law and Kelton use an equally spread design of 36 input combinations. They simulate five replicates for each of the 36 inputs. Based on these 180 I/O data, they fit a second-order polynomial regression model for the average monthly total costs R. They compare this model's predictions with the 'true' $E(R)$ estimated from 10 replicates for each of 420 new and old combinations. Van Beers and Kleijnen, however, use the CSD to select 36 input combinations. They simulate each of their 36 inputs five times and fit a Kriging model to 36 I/O data (implying 36 average outputs). They compare the Kriging predictions for the 420 inputs from the test set with the 420 'true' outputs. They find that the Kriging model gives substantial better *CEIMSE* and *EMax-IMSE*—defined in (8) and (6)—than the regression model.

Figure 6 shows a CSD for 16 input combinations for Law and Kelton's (s, S) example. Note that the procedure selects again more input combinations in the sub-area where the metamodel shows steep slopes.

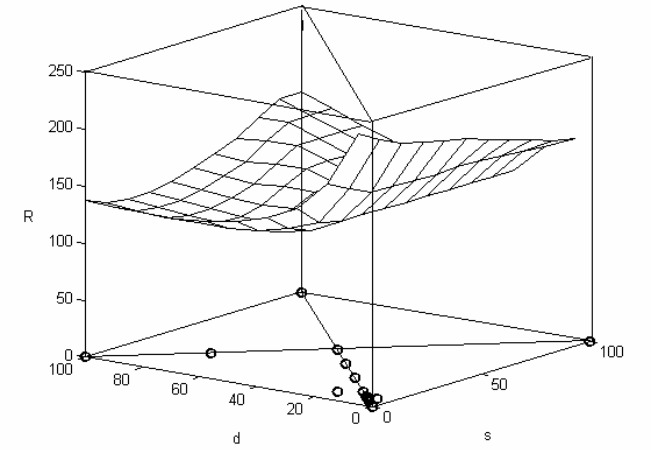

Figure 6: I/O Simulation Data for (s, S) Inventory Model with 16 Scenarios Denoted by ○

4 CONCLUSIONS AND FUTURE RESREARCH

For expensive simulation, it is important to find an efficient design for the experiments with the simulation model. Classic standard designs—such as 2^{k-p} or LHS designs—are general designs that do not account for the characteristics of the input/output (I/O) function that is implied by the simulation model at hand. As an alternative design a *Customized Sequential Design* (CSD) for metamodeling in simulation is derived. The CSD is sequential, because in general sequential procedures are more 'efficient' than fixed-sample procedures; tests confirmed that property. Moreover, the method generates a design that is specific for the given simulation model: it is customized (tailor-made). For deterministic simulation, this customization is achieved through *cross-validation* and *jackknifing*—which are two general statistical techniques. For that simulation type, the method is tested through two academic applications, namely a hyperbolic I/O function and a fourth degree polynomial. For random simulation experiments, the customization uses *bootstrapping*—which is also a general statistical technique (related to jackknifing). The procedure is tested for this simulation type through two classic Operations Research/Management Science (OR/MS) applications, namely the M/M/1 queueing model and an (s, S) inventory management model. Tests showed that for both deterministic simulation and random simulation, customized designs performed better than the classic LHS designs with the same sample size. An interesting property of our procedure is that it simulates relatively many input combinations in those sub-areas that have interesting I/O behavior.

The main conclusions are summarized as follows:

- Kriging metamodels give more accurate predictions than low-order polynomial regression models do,
- Customized Sequential Designs for Kriging metamodels give smaller prediction errors than standard one-shot LHS designs of the same size.

REFERENCES

Bates, R.A., R.J. Buck, E. Riccomagno and H.P. Wynn (1996), Experimental design and observation for large systems. *Royal Statistical Society* 58 (1):7-94.

Campolongo, F., J.P.C. Kleijnen, and T. Andres (2000), Screening methods. In: *Sensitivity analysis*, edited by A. Saltelli, K. Chan, and E.M. Scott, Wiley, Chichester (England), 65-89.

Cressie, N.A.C. (1993). *Statistics for spatial Data*, New York: Wiley.

Ghosh, B.K. and P.K. Sen (editors), 1991, *Handbook of sequential analysis*. Marcel Dekker, Inc., New York.

De Geest, J., T. Dhaene, N. Fache, and D. De Zutter (1999), Adaptive CAD-model building algorithm for general planar microwave structures. *IEEE Transactions on Microwave Theory and Techniques*, 47 (9): 1801-1809.

Efron B. and R.J. Tibshirani (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.

Journel, A.G. and Huijbregts, C.J. (1978), *Mining Geostatistics*. Academic Press, London.

Kleijnen, J.P.C. (1998), Experimental design for sensitivity analysis, optimization, and validation of simulation

models. *Handbook of Simulation*, edited by J. Banks, 173-223, New York: Wiley.

Kleijnen, J.P.C. and J.Helton (1999), Statistical analyses of scatter plots to identify important factors in large-scale simulations. *Reliability Engineering and Systems Safety,* 65 (2): 147-197.

Kleijnen, J.P.C. S.M. Sanchez, T.W. Lucas, and T.M. Cioppa (2004), A user's guide to the brave new world of designing simulation experiments. Working Paper. Available via http://center.kub.nl/staff/kleijnen/papers.html/ [accessed June 24, 2005]

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.

Kleijnen, J.P.C. and D. Deflandre (2005), Validation of regression metamodels in simulation: Bootstrap approach. *European Journal of Operational Research* (in press)

Koehler, J.R. and A.B. Owen (1996), Computer experiments. In *Handbook of statistics,* (13), edited by S. Ghosh, C.R. Rao, 261-308. Amsterdam: Elsevier.

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

Matheron, G. (1962), Traité de géostatistique appliquée. *Memoires du Bureau de Recherches Geologiques et Minieres*, Editions Technip, Paris, 14: 57-59.

McKay, M.D., R.J. Beckman, and W.J. Conover (1979), A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21 (2): 239-245 (reprinted in 2000: *Technometrics* 42 (1): 55-61).

Park, S., J.W. Fowler, G.T. Mackulak, J.B. Keats, and W.M. Carlyle (2002), D-optimal sequential experiments for generating a simulation-based cycle time-throughput curve. *Operations Research* 50 (6): 981-990.

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

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

Sasena, M.J, P. Papalambros, and P. Goovaerts (2002), Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization* 34 (3): 263-278.

Schonlau, M. (1997), *Computer experiments and global optimization*. Doctoral dissertation, Department of Statistics, University of Waterloo.

Van Beers, W.C.M. and J.P.C. Kleijnen (2003), Kriging for interpolation in random simulation. *Journal of the Operational Research Society* 54: 255-262.

Van Beers, W.C.M. and J.P.C. Kleijnen (2004), Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. Available via http://center.kub.nl/staff/kleijnen/papers.html [accessed June 24, 2005]

Van Groenendaal, W.J.H. and J.P.C. Kleijnen (1997), On the assessment of economic risk: factorial design versus Monte Carlo methods. *Reliability Engineering and Systems Safety,* 57: 91-102.

Van Groenigen, J.W. (2000), The influence of variogram parameters on optimal sampling schemes for mapping by Kriging. *Geoderma*, 97: 223-236.

Ye, K.Q. (1998), Orthogonal column Latin hypercubes and their application in computer experiments. *Journal Association Statistical Analysis, Theory and Methods* (93): 1430-1439.

AUTHOR BIOGRAPHY

WIM VAN BEERS is a researcher in the Department of Information Systems and Management of Tilburg University. His research topic is Kriging in simulation. His e-mail address is wvbeers@uvt.nl