Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

TUTORIAL: SIMULATION METAMODELING

Russell R. Barton

The Pennsylvania State University
Smeal College of Business
210 Business Building
University Park, PA 16802 USA

ABSTRACT

The concept of a metamodel has been an important tool for simulation analysis for forty years. These models of simulation models have the advantage of faster execution, and they can (sometimes) provide insight on the nature of the simulation response as a function of design and input distribution parameters. This introductory tutorial will describe metamodeling uses and associated processes, survey commonly used metamodel types and associated experiment designs, and give a brief description of some recent developments and how they may affect future "mainstream" simulation metamodeling.

1 INTRODUCTION

Discrete-event simulation models allow users to examine in relatively high fidelity the expected behavior of manufacturing, transportation, health care or other systems. The simulation model can be exercised at lower cost and with less risk than the real system. Further, the real system may not yet exist, so simulation provides performance prediction for the as-yet-to-be-constructed real system. Exercising such a model in an ad hoc way may not lead to general insights on the behavior of the real (either extant or imagined) system being modeled, however. Further, when the simulation model fidelity is high and the system is complex, the computational effort to simulate can be substantial. Simulationists then often fit a computationally efficient approximation to the simulation model: a metamodel. In this introductory tutorial we examine the role of metamodels and the metamodeling activity in simulation studies. Metamodeling permits special insights into system performance, and enables extensive exercise of the system under different design conditions at relatively modest computational cost.

This is the first WSC introductory tutorial focused on metamodeling in at least seventeen years (Barton 1998). Closely related tutorials on the design of simulation experiments appear regularly, however. See for example the tutorials by Law (2014), Sanchez and Wan (2012), Barton (2010), and Kleijnen (2008). This is not to say that the field is not active: there have been many papers on new metamodeling methods and on new uses for simulation metamodels. At WSC'14 alone there were two sessions dedicated to metamodeling, and a total of ten papers across all sessions. And see the very good advanced tutorial by Staum (2009).

This tutorial will provide an accessible view to metamodeling for the beginner. For that reason, open access materials receive preference: a reference to a *Management Science* paper may be omitted in favor of an earlier WSC publication - these can be freely accessed under the Archive link at the WSC home page (www.wintersim.org). Advanced methods will not be the focus. While this tutorial will discuss the design of simulation experiments, the emphasis will be on the historical context of metamodeling, metamodel types, and the ultimate uses of metamodels, with guidance on the overall metamodeling process.

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

2 METAMODELING: BASICS, PURPOSE, AND PROCESSES

2.1 Basics of Metamodeling

Metamodels approximate the input-output behavior of simulation models. The term metamodel was popularized and developed by Jack Kleijnen (for example, Kleijnen 1975), but the term and concept were both originated by Robert Blanning (1974, 1975a, 1975b). The term indicates a mathematical approximation that *models* the behavior of another *model*. These approximations are also called surrogate and response surface models. Early work explored response surface models for simulations, under the name of "regression" or "experimental designs" (Walsh 1963, Burdick and Naylor 1966, Hunter and Naylor 1970). While these focused on stochastic, typically discrete-event, simulation (as are ours), metamodels are also used to approximate other kinds of simulation models (e.g., finite element, circuit, stochastic flow, and boundary element simulation models). An early application of response surface modeling in water resource planning simulation appears in Hufschmidt and Fiering (1966). Hufschmidt is also a coauthor in Maass et al. (1962) which has a section on response surface modeling of a water system simulation model.

A metamodel is a function, say f, that takes some simulation model design parameters as inputs, represented here by a vector x, and produces an approximation to some characteristic of a simulation output, say $g(Y)$ (e.g., mean of Y, standard deviation of Y, 0.9-quantile of Y, etc., of some performance measure). Examples of model design parameters include input probability distribution parameters, such as arrival rate and mean service time; and system configuration parameters, such as the number of servers, service priority, operational protocols, and buffer capacity. For now, assume that any design parameter can be coded numerically, even if only as a 0-1 variable. There will be more about how to do this later in the tutorial. Examples of simulation outputs are time in the system for a set of jobs or customers, utilization of a particular resource (e.g., operator, machine), or perhaps net revenue over a specific time period. Generally these outputs are averaged over the length of a simulation run, but vary randomly from run to run. If the value of the characteristic is $Y(x)$ for an actual simulation run with the design parameters set to the values in x, then we represent the fitted metamodel approximation $f(x)$, as:

$$h(Y(x)) \approx f(x), \tag{1}$$

where h represents some function of the random variable Y. Often it is a composition of functions: some nonlinear function of the expected value of Y or of a quantile of or of a quantile of Y. Note that the distribution of y depends on the values of the design parameters. The simplest and most common metamodel type is linear regression. The term "linear" refers to the way the unknown coefficients come into the model. Linear regression can capture curvilinear relationships.

To illustrate the basics of metamodeling, we will fit two regression metamodels to data from a simple queueing simulation. Although simulation is not needed for an M/M/1 queue, such a system is easy to understand and has behavior that is frequently seen in simulations. In this case we want to explore how the average waiting time (the *output*) varies with service speed (the *input*, or *design parameter*), assuming an arrival rate of 1 customer per unit time (in some scaled units of time). The plot below, made using Minitab® (Minitab 2015) shows the results of simulations of 5000 customers in systems with mean service times of 0.7, 0.75. 0.8, 0.85, 0.9 and 0.95.

If we fit a linear regression model to this data, the metamodel produces the fit seen in Figure 2. If we fit a quadratic metamodel to these data it produces the fit shown in Figure 3. Clearly, there are problems with the fidelity of both of these fitted metamodels. We will return to this example throughout the tutorial to discuss each aspect of the metamodeling process.

For Figure 2, the metamodel corresponding to (1) has $Y \equiv$ average waiting time for the 5000 customers in a simulation run; $h(Y) \equiv$ the expected value of this run average, i.e., $\mathrm{E}(Y)$; $x \equiv$ mean service time (the x vector has only one component); and $f(x) = -44.19 + 61.42x$. For Figure 3, the fitted metamodel is $f(x) = 269.9 - 708.3x + 466.5x^2$.

Now that we have a basic understanding of what a metamodel is, we can discuss purposes for which metamodels are built.

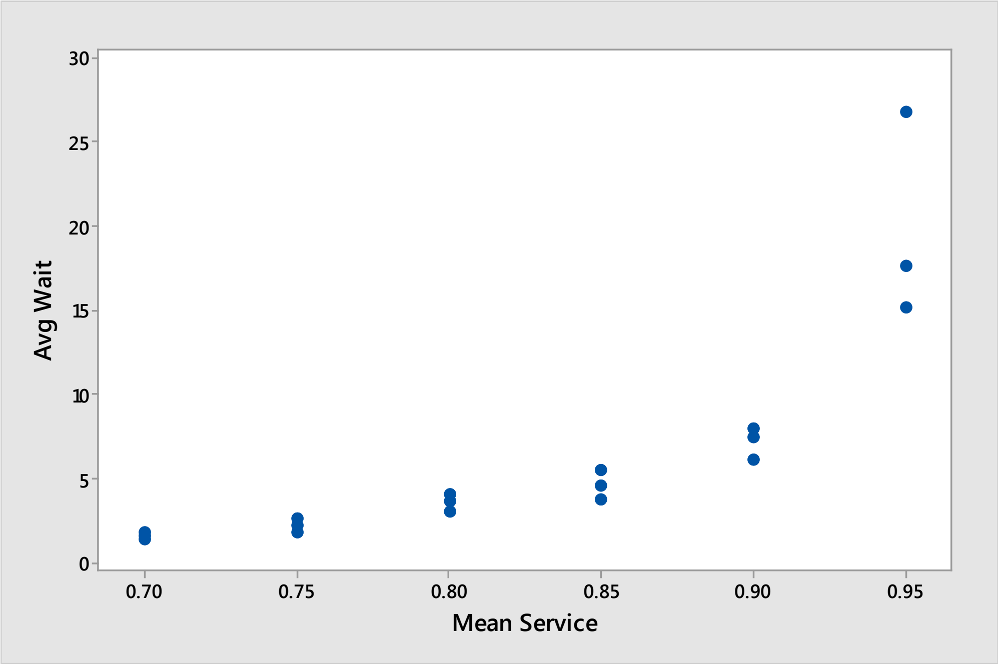

Figure 1: Output of M/M/1 experiments, three replications each at six mean service time settings.

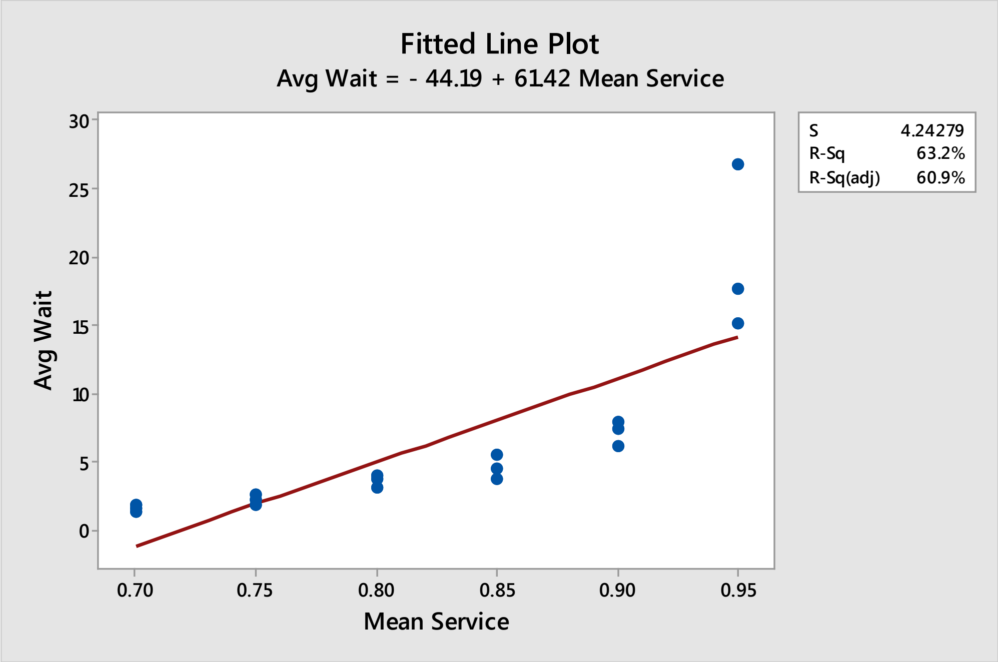

Figure 2: A regression metamodel fitted to the data, with intercept and linear term only.

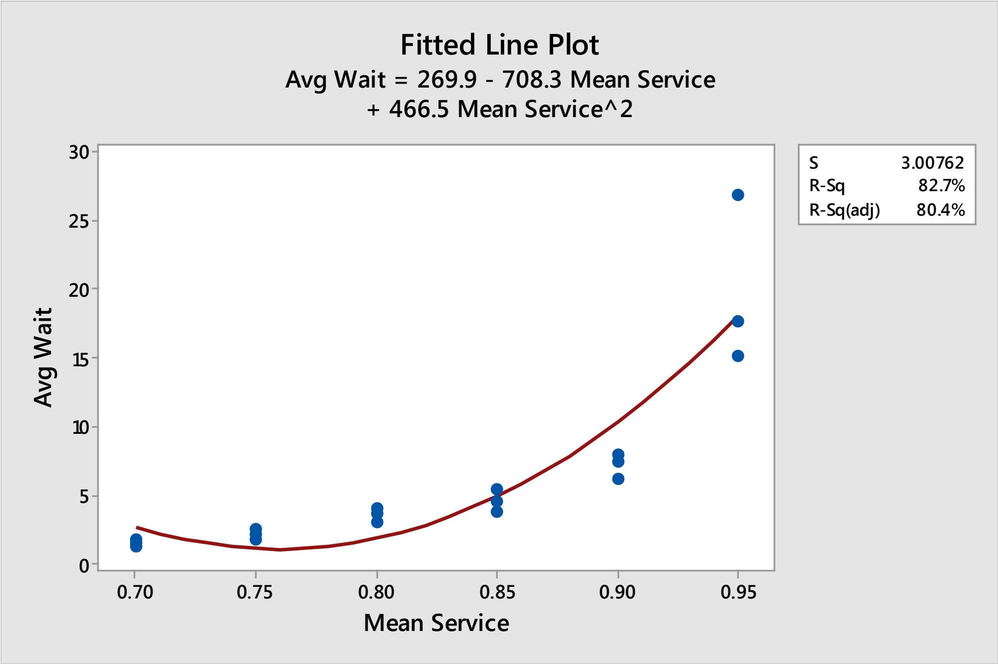

Figure 3: The fitted quadratic regression metamodel is an improvement.

2.2 Metamodeling Purpose

Metamodels generally have three characteristics that give them advantage over the simulation model for certain purposes. A metamodel *f* generally has *explicit form*, *deterministic output*, and, once fitted, is *computationally inexpensive* to evaluate.

Due to their explicit form, metamodels are often described as providing *insight* - that is, an understanding of the general relationship between design parameters and some performance measure related to the simulation output. For the example in Figure 2, we find a positive coefficient, which indicates both a *positive relationship* between service time and the expected value of average waiting time, and the magnitude of the slope coefficient gives the level of *sensitivity*. For a metamodel with one design parameter this sensitivity measure may not be very interesting, but when there are tens or hundreds of design parameters, the size of the coefficients can be used to identify the most influential design variables and *screen* the others. The coefficients of the metamodel shown in Figure 3 provide more insight: the relationship between waiting time and service time is *convex*. We know that because the coefficient of the squared term is positive. Further, it is easy to evaluate either of these metamodels for many values of service time, allowing one to characterize global variation in the mean value.

Metamodels, once fitted, can be used as a proxy, to evaluate instead of making (computationally expensive and stochastic) simulation runs. Further, because of their explicit form, they can be used in many computationally intensive operations, such as optimization (Cheng and Currie 2004, Barton 2009), input model uncertainty (Xie, Nelson and Barton 2014), quantiles and conditional value at risk (Chen and Kim 2013) and robust design (Dellino, Kleijnen and Meloni 2009).

2.3 The Metamodeling Process

The rest of this introductory tutorial is organized around the steps of the metamodeling process. A nine-step process was described by Burdick and Naylor (1966), and other authors have provided alternative process descriptions (see Section 9: FOR FURTHER STUDY below). We will follow a seven-step process.

Table 1: The metamodeling process.

Step	Activity
1	Determine Purpose(s) for Metamodeling
2	Identify Design Parameter(s) and Output(s)
3	Choose Metamodel Type
4	Based on Metamodel Type and on Purpose, Choose Experiment Design to Fit Metamodel
5	Conduct Simulation Runs Specified by the Experiment Design; Fit Metamodel
6	Validate Metamodel Adequacy: If Unsatisfactory (usually) Return to Step 3
7	Use Metamodel for Intended Purposes

3 IDENTIFYING DESIGN PARAMETERS AND OUTPUTS

3.1 Design Parameter and Output Selection

Typically, simulation models have many design parameters that might be included in a metamodel. Each parametric probability distribution used in the model has parameters. These are called input distribution parameters. They characterize randomly varying interarrival times, service times, times until machine breakdown, message lengths, routing choices, transport times, number of workers out sick, uncontrollable environmental factors, and other characteristics of the simulated system that have a stochastic nature.

In addition, there are design parameters that are deterministic in value: number of workers scheduled to work at a particular time, number of machines, processing protocols and other characteristics of the system being modeled that can be changed, either for a currently operating system or some system to be built in the future.

The particular parameters to include in the metamodel depend on three things: i) a desire to include parameters that the decision maker would like to explore changing; ii) the need to model the impact of any uncontrollable environmental factors that affect system performance; and iii) the recognition that more metamodel parameters generally means a larger fitting experiment, and more computational effort to fit the metamodel. Further, for regression metamodels, the inclusion of higher order terms to capture interaction effects and nonlinearity (x^2 for the M/M/1 example) also adds to the size of the experiment design. Cause-effect diagrams and a-priori plots can help identify the design parameters and any expected interactions or nonlinearity (Barton 2010). For the M/M/1 example above, the experiment design was presented as given. In actuality it would be selected after considering whether to include x^2 in the metamodel.

Functions of the simulation outputs that are of interest generally are each modeled separately, that is, a separate metamodel is fitted for each function of the simulation outputs needed for the purpose(s) identified in Step 1. Multiple response surface models were considered in the early paper by Burdick and Naylor (1966) and many subsequent authors. True multivariate metamodels have been used for physics-based simulations (Tu 2003). For the M/M/1 example, only one function of one simulation output is of interest: h corresponds to the expected value, and Y is the average waiting time of 5000 simulated customers. Occasionally, h will be chosen to reduce the nonlinearity of the output with respect to the design parameters, or to achieve equal (homogeneous) output variance across the design space. Methods for choosing such transformations are summarized in Barton and Meckesheimer (2006).

3.2 Continuous and Discrete Design Parameters and Outputs

Generally metamodeling assumes that all design parameters and outputs can take on continuously varying values. But many design parameters are discrete. Examples include numbers of servers, machines or other resources, buffer sizes, number of products, type of processing protocol, system configuration alternatives, and so forth.

When the parameter has a numerical value that is restricted to a discrete set of values, then assuming that the parameter can take on a continuous set of values is practical. The discrete nature places a restriction on the experiment design that is used for fitting, but the fitted metamodel can be evaluated on the discrete set of allowed values.

When there are only two values for the parameter (e.g., *Protocol A* and *Protocol B*), a discrete numerical characterization can be assigned, for example, $x = 0$ for *Protocol A* and $x = 1$ for *Protocol B*. If the parameter does not have numerical value and there are more than two levels, numerical conversion is still possible, but requires additional x components. For example, for three protocols, let $x_1 = 1$ if *Protocol A* is used, and = 0 otherwise. Let $x_2 = 1$ if *Protocol B* is used, and = 0 otherwise. If *Protocol C* is used, then both x_1 and $x_2 = 0$. Then metamodel terms for x_1 (or x_2) will indicate the differences of *Protocol A* (or *Protocol B*) from *Protocol C*.

When the simulation output is discrete (e.g., success or failure), there is a restriction on the type of metamodel. These are generally called classifier or discriminant metamodels. Meckesheimer et al. (2001) consider metamodels with both continuous and discrete responses.

3.3 Scaling/Coding Parameters and Outputs

In addition to the output transformations and qualitative variable coding mentioned previously, the ability of the metamodel to provide insight depends on careful scaling and coding of all design parameters. Generally, code all numerical design parameters so that -1 is the smallest value taken, and +1 is the largest value. This coding is accomplished by the following:

$$x_{new} = 2[x - ((x_{max} + x_{min})/2)/(x_{max} - x_{min})]. \tag{2}$$

Compare the insight from the fitted metamodel in Figure 2 with that in Figure 4, based on the same data but rescaling the average service time using (2), to -1 for .7 and +1 for .95.

Insight from Figure 2: a unit increase in mean service time would result in an increase in average waiting time of approximately 61 time units (note: a unit increase in mean service time could not happen – the system would be unstable). Also, if the mean service time were reduced toward zero, the average waiting time would tend toward -44 time units (note: negative time is impossible).

Insight from Figure 4: moving from the middle service time (.825) to the highest service time (.95) will increase the average waiting time by approximately 7.7 time units. If the system is operated at the middle service time value (.825), the average waiting time will be approximately 6.5 time units.

Clearly, basic insights from the regression model coefficients fail to materialize without careful coding of the design parameter value(s).

Further, Figure 5 shows that, unlike the difference between the models in Figures 2 and 3, the linear coefficient does not change between Figures 4 and 5. Note that the metamodel fit does not change – Figure 2 has the same shape as Figure 4 and Figure 3 as Figure 5 – only the interpretation of the coefficients changes. The intercept did change between Figure 4 and Figure 5, however. Fixing this requires more than coding of the design parameters, it requires coding the functions of those parameters used in the regression models. The topic of orthogonal polynomials is beyond this tutorial. See for example Montgomery (2012).

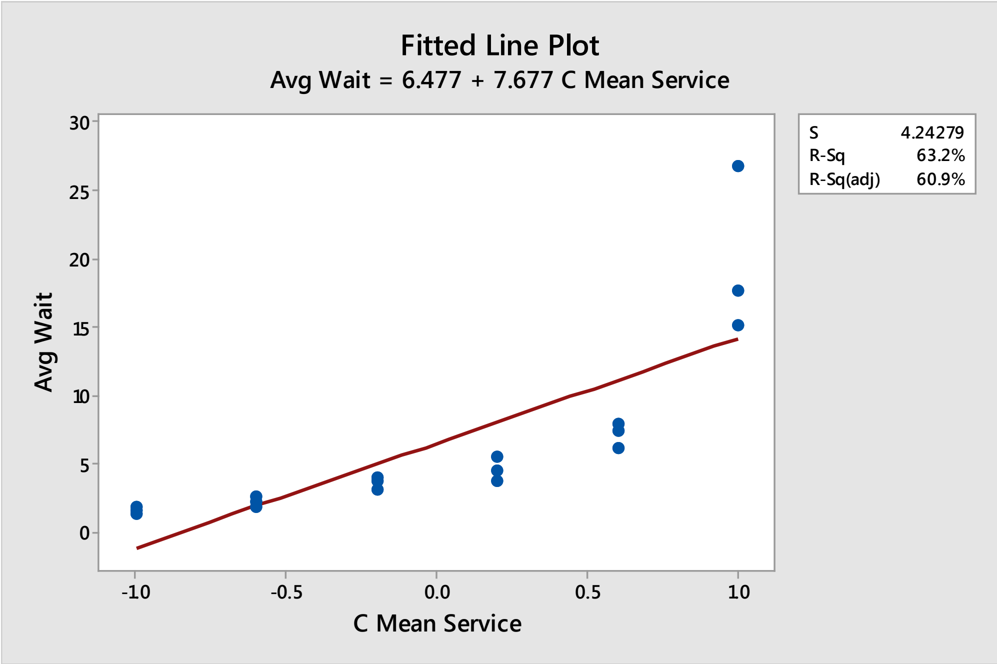

Figure 4: The linear regression metamodel with coded service time: fit is identical to that in Figure 2.

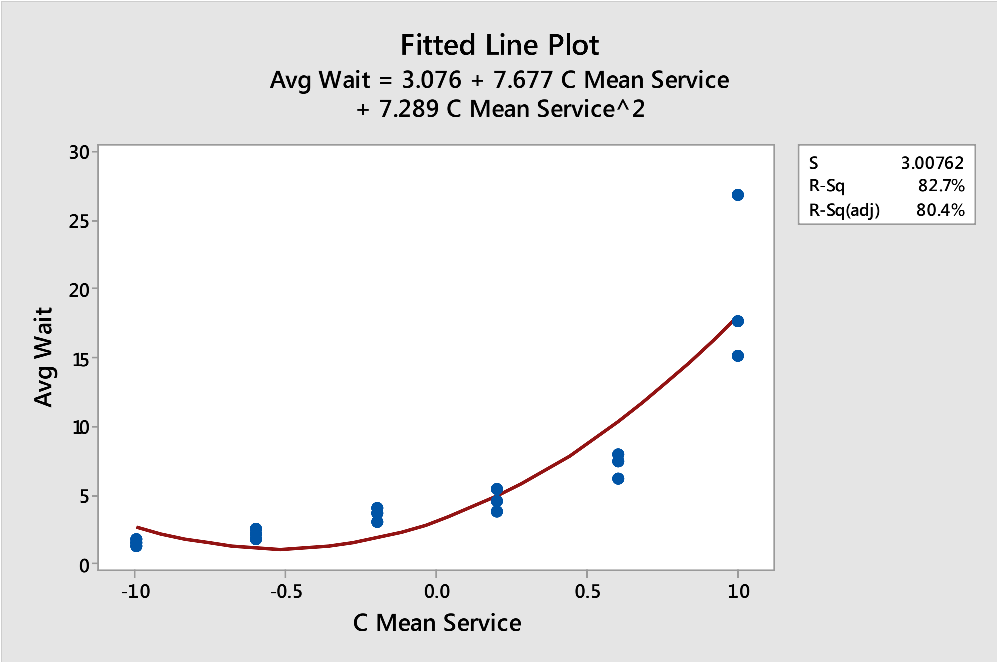

Figure 5: The quadratic regression metamodel with coded service time.

4 CHOOSE A METAMODEL TYPE

For this introductory tutorial we focus on two types of metamodels: linear regression and stochastic Kriging (spatial correlation) models. There are descriptions of other metamodel types in Barton (1998) and Barton (2009). We identify the metamodel type before selecting an experiment design for practical reasons: factorial and fractional-factorial designs, appropriate for linear regression, can cause significant numerical difficulties when used to fit Kriging models. Further, the complexity of the proposed regression model places (minimum) requirements on the type of factorial or fractional-factorial design that can be employed for fitting.

4.1 Linear Regression Metamodels

Regression metamodels use a probability model to characterize the simulation output of interest. The form of the probability model that characterizes the simulation output is:

$$Y = \beta_0 + \beta_1 g_1(x_1, x_2, ..., x_d) + ... + \beta_p g_p(x_1, x_2, ..., x_d) + \varepsilon, \tag{3}$$

where ε are independent, normal random quantities with mean zero and unknown variance and there are d design parameters. Again, the term *linear* comes from the fact that the unknown coefficients (β's) to be fitted in the experiment appear linearly (as multipliers) in the model. The g functions can be nonlinear in the x's. There are p terms in the model (not counting the intercept), allowing for terms involving functions of one or more design parameters, for example, $g_5(x_1, x_2, ..., x_d) = x_1^2$, or $g_{12}(x_1, x_2, ..., x_d) = x_1x_5$. The assumption is that the variance does not change depending on the values of $(x_1, x_2, ..., x_d)$. This model (3) implies

$$\mathrm{E}(Y) = \beta_0 + \beta_1 g_1(x_1, x_2, ..., x_d) + ... + \beta_p g_p(x_1, x_2, ..., x_d), \tag{4}$$

and the metamodel $f(x)$ will match (4) but with estimated values b_0, b_1, …, b_p for the unknown coefficients. Since the estimated values b_0, b_1, …, b_p will vary randomly from one experiment to the next, conceptually the fitted metamodel depends randomly on the data. Given a set of data $\{x_i, y_i\}$ where $x_i = (x_{i1}, x_{i2}, …, x_{id})$ is the vector of design parameter values for the i^{th} simulation run, and y_i is the corresponding output, let X be the matrix whose i^{th} row is x_i and let y be the column vector consisting of the elements $\{y_i\}$. Then the estimated coefficient vector $b = (b_0, b_1, …b_p)$ is computed by:

$$b = (X'X)^{-1}X'y. \tag{5}$$

where the prime symbol denotes matrix transpose. The solution (5) minimizes the average squared deviation of the metamodel prediction from the observed simulation outputs (MSE):

$$\min \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2 / n \text{, where the minimization is over } b \text{ and } \hat{Y}_i = \sum_{j=0}^{p} b_j g_j(x_{i1}, x_{i2}, ..., x_{ip}). \tag{6}$$

The fitted regression metamodel is then

$$\hat{Y}(x) = \sum_{j=0}^{p} b_j g_j(x_1, x_2, ..., x_p). \tag{7}$$

The intercept term b_0 has corresponding $g_0 \equiv 1$. For the M/M/1 example, the fitted metamodel in Figure 4 has $b_0 = 6.477$, and $b_1 = 7.677$. The fitted metamodel in Figure 5 has $b_0 = 3.076$, $b_1 = 7.677$, and $b_2 = 7.289$.

Linear regression models have simple form and so provide direct insight on the behavior of the simulation. When design parameters are coded over [-1, +1], then the magnitude of the linear coefficients indicate the relative sensitivities of the simulation output to all design parameters (over the defined ranges of parameter values). Similarly, quadratic coefficients can indicate nonlinearity, and convexity/concavity. Coefficients for cross-product terms indicate interaction effects – the sensitivity of the output to changes in one design parameter may vary, depending on the setting of another design parameter. Also, linear regression models are readily available in commercial statistical software.

Linear regression models using polynomial functions of the design parameters have limited flexibility, however. Figure 6 shows an attempt to find a better-fitting metamodel for the M/M/1 example by adding a cubic term. While the curve comes closer to the observed waiting times overall, it is no longer monotonically increasing, something we expect in a metamodel of mean waiting time versus mean service time. Adding more terms improves the fit near the six experiment design points but increases the excursions of the metamodel away from the design points. For the classic illustration of this "excursions" shortcoming of polynomial models see Figure 1 in Barton (1992).

Looking at Figures 1-6 it is apparent that the variance of the response is larger at higher average waiting times. This is particularly important for models where the output is some function of a queueing system, as is often the case in discrete-event simulation. One approach to reducing this *heteroscedasticity* is to take different numbers of replications at different design points. By taking more replications at high-variance points, the variance of the average response at such points is reduced. This is an expensive proposition though: the spread you see is related to the standard deviation, which is only reduced as the square root, so to reduce the spread by a factor of two requires 4x the replications!

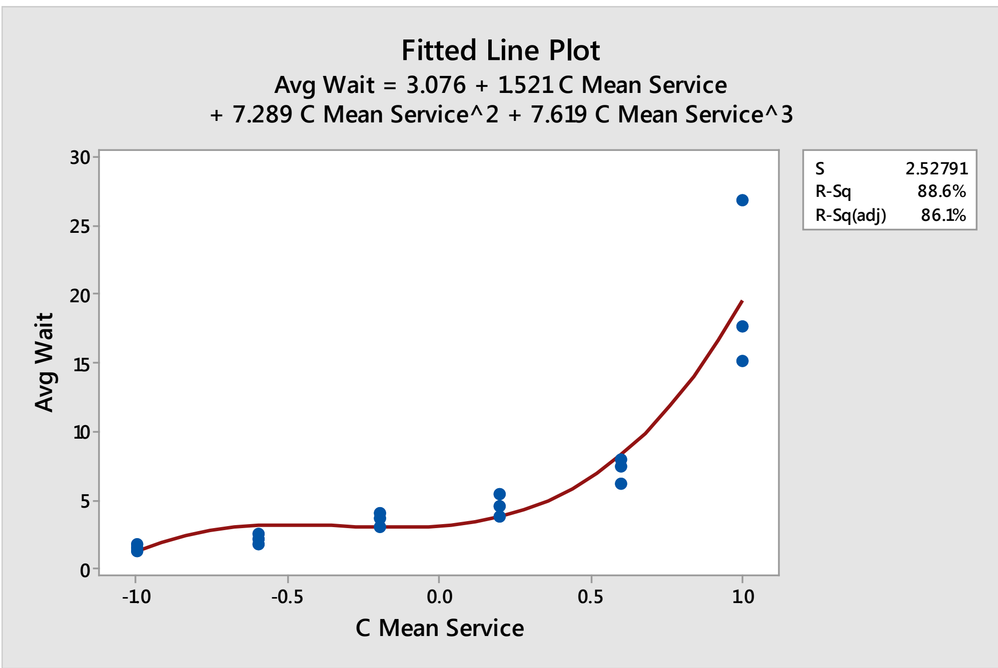

Figure 6: A cubic regression metamodel is not monotonic.

Often problems with heteroscedasticity *and* fitting can be reduced by transforming the dependent variable(s). A typical transformation for queueing output data is the logarithmic transformation. Figure 7 shows the same model as Figure 5 but using ln(Y) as the dependent variable. Note that i) the large differences in spread across the design points is reduced and ii) the fit as measured by R^2 is better.

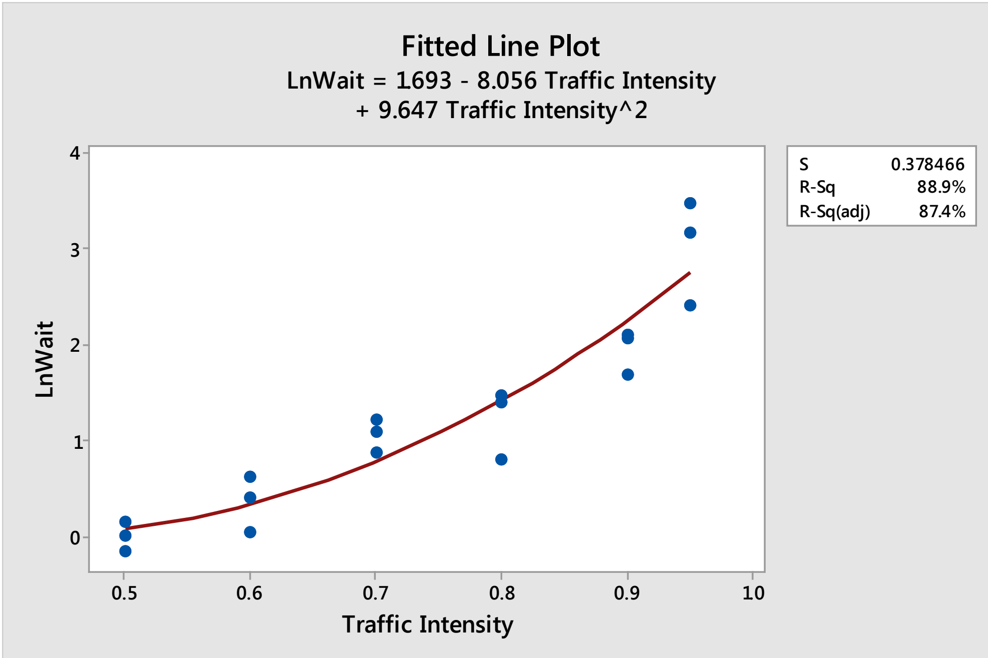

Figure 7: Transforming to ln(Avg Wait) reduces heteroscedasticity and improves fit.

4.2 Stochastic Kriging Metamodel

Stochastic Kriging methods for discrete-event simulation metamodeling are a relatively recent development (Ankenman, Nelson and Staum, 2008). The simplest Kriging probability model is:

$$Y(x) = \beta_0 + M(x), \tag{8}$$

where M is the realization of a mean zero random field. That means it is a function drawn at random from the set of all functions whose nearby values are correlated according to a prespecified spatial correlation function. For that reason these models are also called spatial correlation models. Other g terms as in (4) can be added to the model, but are rarely necessary for a good fit. For a history of spatial correlation models and associated references, see Barton (1998).

Such models have been used to approximate deterministic response functions, since once the realization occurs, the model (8) has no intrinsic randomness. Since stochastic simulation models have an output with intrinsic randomness, Nelson, Ankenman and Staum added a normally distributed intrinsic error term, $\varepsilon(x)$ to (8), and allow covariance between $\varepsilon(x_i)$ and $\varepsilon(x_j)$. In this metamodel i indexes a unique set of design parameter values, $i = 1, \ldots, k$, with n_i replications run with the design parameter vector set to x_i. The fitted stochastic Kriging metamodel is:

$$\hat{Y}(x) = b_0 + [t^2 R_M(\hat{\theta}) + \hat{\Sigma}_\varepsilon]^{-1}(\bar{Y} - b_0 1_k) \tag{9}$$

where t^2 and $\hat{\theta}$ are spatial correlation parameters estimated from the experimental data, $\hat{\Sigma}_\varepsilon$ is the sample intrinsic error covariance matrix (sample variances and covariances across all experiment design points), $R_M(\hat{\theta})$ is an approximate spatial correlation matrix computed using $\hat{\theta}$, and 1_k is a k-dimensional vector of ones.

Stochastic Kriging models have great flexibility. They can model more complex response function shapes than is possible with polynomial regression metamodels. If one requires a global approximation to a nonlinear response, it is unlikely that regression will provide a good fit, however. In that case, stochastic Kriging provides an alternative. This makes them very attractive, but comes at a cost. First the model is more complex to fit, and the fitting and prediction software for (9) is not commonly a part of simulation or statistical packages. A package of MATLAB routines implementing stochastic Kriging is available on the Web at http://www.stochasticKriging.net/. Second, fitted model coefficients give some indication of how rapidly the response changes as components of *x* change, but the detailed insight available in a fitted regression model cannot be obtained. Further, if experimental run conditions are scarce, predictions in design parameter space between experimental runs can be significantly in error due to mean reversion. Staum (2009) shows some error patterns that can occur. Because this is an introductory tutorial, the best advice here is to stay tuned – it is likely that stochastic Kriging metamodels are in your future.

4.3 Choosing a Metamodel Type

In addition to stochastic Kriging, there are many other metamodel types to choose from, including radial basis functions, neural networks, and regression trees. See Chen et al. 2006 for a review. Because of the simplicity, broad availability of software, and advantage in terms of insight, linear regression seems the place to start in metamodeling.

5 CHOOSE EXPERIMENT DESIGN

In choosing an experiment design, one determines the number of distinct simulation settings to be run, and the specific values of the design parameters for each of these runs. There are many strategies for selecting the number of runs and the factor settings for each run. These include random designs, optimal designs, combinatorial designs, Latin hypercube designs, orthogonal arrays, uniform designs, mixture designs, sequential designs, and factorial designs. For regression metamodels, the number and kinds of terms to be fitted places constraints on the minimum number of runs and minimum number of levels tested for each design parameter. Barton (2010) is a reference for this discussion, but see also Sanchez and Wan (2012).

5.1 Experiment Designs for Regression

Experiment designs for regression are well-developed. There is a clear link between the form of the model being fitted and the kind of experiment design that is preferred. Typically, regression designs are either factorial designs or fractional factorial designs. Factorial designs are based on a grid, with each factor tested in combination with every level of every other factor. Factorial designs are attractive for three reasons: i) the number of levels that are required for each factor is one greater than the highest-order power of that variable in the model, and the resulting design permits the estimation of coefficients for all cross-product terms ii) they are probably the most commonly used class of designs, and iii) the resulting set of run conditions are easy to visualize graphically for as many as nine factors.

The disadvantage of factorial designs is that they require a large number of distinct runs when the number of factors and/or the number of levels of the factors are large. In this case, fractional-factorials are often employed. See Sanchez and Wan (2012) for a good overview. Table 2 gives some guidance on experiment designs appropriate for regression modeling, depending on the purpose and nature of the model.

5.2 Experiment Designs for Stochastic Kriging

While the focus of this tutorial is on regression metamodels, it is instructive to see how the strategy would differ for a different metamodel type. Very little research has been published on experiment designs for stochastic Kriging (but see Xie, Ankenman, and Nelson 2010 for a study using common random numbers). Factorial designs have been found to work poorly with deterministic Kriging models generally. Latin hypercube designs are often used for Kriging models, but the set of Latin hypercube designs has many designs (for the same number of levels and runs) that give poor coverage. For that reason it is important to sample many Latin hypercube designs and choose one with good properties. For example, one can choose the design that maximizes the minimum distance between any two design parameter vectors (maximin). Good alternatives are Hammersley sampling sequences, orthogonal arrays, and uniform designs. Alternative designs are discussed in Chen et al. (2006), Sanchez and Wan (2012), and Kleijnen (2015).

Table 2: Experiment designs for regression metamodels.

Objective	Minimum Size Factorial Designs	References
Initial screening	Saturated and supersaturated fractional factorial	Li and Lin (2003)
Sensitivity	Saturated and resolution III Plackett-Burman fractional factorial	Kleijnen (2015)
Insight	3-level full or fractional factorial or central composite (more than 3 levels needed to check lack of fit)	Sanchez and Sanchez (2005), Kleijnen (2015), Montgomery (2012), Sanchez and Wan (2012)
Optimization	3- or more level fractional factorial	Montgomery (2012), Law (2015), Kleijnen (2015)

6 CONDUCT RUNS AND FIT MODEL

Unlike physical experiments, external environmental factors generally do not affect simulation results. Once the model type and design selection steps are complete, running the experimental conditions is straightforward, provided the experimenter keeps the simulation results attached to the correct values of the design parameter settings for each run.

It is possible to deliberately introduce correlation of randomness across runs with different design parameter settings. This can be done by reusing random number streams for runs with different design paramemeter settings. This correlation induction can be hard to achieve, but it can result in better fits for regression metamodels (Schruben 1979, Tew and Wilson 1987).

The work by Chen, Ankenman and Nelson (2010) that was mentioned above found that correlation induction using common random numbers was not effective for stochastic Kriging.

6.1 Incorporating Gradient Information in Fitting the Metamodel

It is often possible to generate more than just an output value at the end of a simulation run. When one can also estimate the gradient of the response with respect to the design parameters, metamodel fit can be improved, often quite significantly (Qu and Fu 2012).

7 VALIDATING METAMODEL ADEQUACY

The fitted model must be checked to see if the fidelity is adequate for the intended use. For a regression metamodel for screening, simple statistical significance checks may be sufficient. For purposes where fidelity is important, additional goodness of fit tests are employed. There are well-developed goodness of fit tests for regression that appear in all commercial packages. To use them you will need to include more than the minimum number of design parameter levels.

For regression models, mean squared error (MSE) is provided automatically with the fit, as is R^2. High R^2 values can occur when there are just a few extreme values of x and correspondingly high or low values of the response. MSE will give a better assessment of fit in this case.

A general-purpose measure of fit that can be used outside the regression setting is to leave some design configurations and corresponding responses out of the set used to fit the model and then check the error of the fitted model at the design parameter settings left out of the fitting process. This process can be computationally expensive if it is repeated for each possible omission. Meckesheimer et al. (2002) provide some efficient and effective assessment methods of this sort.

8 PUT THE METAMODEL TO USE

Assuming the fitted model passed the validation checks, it is ready to be used. Congratulations on successfully developing a metamodel! Remember though, that uses beyond the original purpose (e.g., using a screening metamodel for prediction or, worse, optimization) are not appropriate.

9 FOR FURTHER STUDY

Many books on simulation have a chapter on the design of experiments, which usually cover metamodeling. Three books with comprehensive coverage are Friedman (1996), Kleijnen (2015), and Law (2015). Kleijnen's book includes design of experiments issues for Kriging metamodels.

ACKNOWLEDGMENTS

I have had the good fortune to benefit from the wisdom of the authors listed below. The Winter Simulation Conference has provided an opportunity to meet many of those listed below for direct discussions, and many others, who have helped my understanding of and appreciation for simulation metamodels. I thank the reviewers, particularly Susan Sanchez, for many helpful suggestions.

REFERENCES

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2010. *Discrete-Event System Simulation*. 5th ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Barton, R. R. 1992. "Metamodels for Simulation Input-Output Relations." In *Proceedings of the 1992 Winter Simulation Conference*, edited by J. J. Swain, D. Goldsman, R. C. Crain and J. R. Wilson, 289-299. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Barton, R. R. 1998. "Simulation Metamodels." In *Proceedings of the 1998 Winter Simulation Conference*, edited by D. J. Medeiros, E. F. Watson, J. S. Carson and M. S. Manivannan, 167-174. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Barton, R. R. 2009. "Simulation Optimization using Metamodels." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 230-238. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Barton, R. R. 2010. "Simulation Experiment Design." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 75-86. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Barton, R. R. and M. Meckesheimer. 2006. "Metamodel-Based Simulation Optimization." Chapter 18 in *Handbook in Operations Research and Management Science Vol. 13: Simulation*, edited by S. G. Henderson and B. L. Nelson, 535-574. Amsterdam: North Holland.

Blanning, R. W. 1974. "The Sources and Uses of Sensitivity Information." *Interfaces* 4:32-38.

Blanning, R. W. 1975. "The Construction and Implementation of Metamodels." *Simulation* 24: 177-184.

Blanning, R. W. 1975. "Response to Michel, Kleijnen and Permut." *Interfaces* 5:24-25.

Burdick, D. S. and T. H. Naylor. 1966. "Design of Computer Simulation Experiments for Industrial Systems." *Communications of the ACM* 9:329-339.

Chen, V. C. P., K.-L. Tsui, R. R. Barton and M. Meckesheimer. 2006. "A Review on Design, Modeling and Applications of Computer Experiments." *IIE Transactions* 38, 273-291.

Chen X., B. E. Ankenman and B. L. Nelson. 2010. "Common Random Numbers and Stochastic Kriging. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 947-956. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Chen, X. and K. Kim. 2013. "Building Metamodels for Quantile-based Measures including Sectioning." In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl. 521-532. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Cheng, R. C. H. and C. S. M. Currie. 2004. "Optimization by Simulation Metamodeling Methods." In *Proceedings of the 2004 Winter Simulation Conference*, edited by R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 485-490. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Dellino, G., J. P. C. Kleijnen and C. Meloni. 2009. "Robust Simulation-Optimization using Metamodels." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 540-550. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Friedman, L. w. 1996. *The Simulation Metamodel*. Norwell, MA: Kluwer.

Hufschmidt, M. M. and M. B. Fiering. 1966. *Simulation Techniques for Design of Water Resource Systems*. Cambridge, MA: Harvard University Press.

Hunter, J. S. and T. H. Naylor. 1970. "Experimental Designs for Computer Experiments." *Management Science* 16:422-434.

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. 2008. "Design of Experiments: Overview." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 479-488. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Kleijnen, J. P. C. 2015. *Design and Analysis of Simulation Experiments.* 2nd ed. New York; London: Springer.

Law, A. W. 2014. "A Tutorial on Design of Experiments for Simulation Modeling." In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 66-80. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Law, A. W. 2015. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill.

Li, R. and D. K. J. Lin. 2003. "Analysis Methods for Supersaturated Design: Some Comparisons." *Journal of Data Science* 1:249-260.

Maass, A., M. M. Hufschmidt, R. Dorfman, H. A. Thomas, Jr., S. A. Marglin and G. M. Fair. 1962. *Design of Water Resource Systems*. Cambridge, MA: Harvard University Press.

Meckesheimer, M., R. R. Barton, T. Simpson, F. Limayem and B. Yannou. 2001. "Metamodeling of Combined Discrete/Continuous Responses." *AIAA Journal* 39:1950-1959.

Meckesheimer, M., A. J. Booker, R. R. Barton and T. W. Simpson. 2002. "Computationally Inexpensive Metamodel Assessment Strategies." *AIAA Journal* 40:2053-2060.

Minitab. 2015. "Minitab 17." Accessed May 7th, 2015. http://www.minitab.com/en-us/.

Montgomery, D. C. 2012. *Design and Analysis of Experiments*. 7th ed. New York: Wiley.

Qu, H. and M. C. Fu. 2012. "On Direct Gradient Enhanced Metamodels." In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 478-489. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M. and P. J. Sanchez. 2005. "Very Large Fractional Factorial and Central Composite Designs." *ACM Transactions on Modeling and Computer Simulation* 15, 362-377.

Sanchez, S. M. and H. Wan. 2012. "Work Smarter, Not Harder: a Tutorial on Designing and Conducting Simulation Experiments." In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1929-1943. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schruben, L. W. 1979. "Designing Correlation Induction Strategies for Simulation Experiments." In *Current Issues in Computer Simulation*, edited by N. R. Adam and A. Dogramaci, 235–256. New York: Academic Press.

Staum. J. 2009. "Better Simulation Metamodeling: The Why, What, and How of Stochastic Kriging." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 119-133. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Tew, J. D. and J. R. Wilson. 1987. "Metamodel Estimation using Integrated Correlation Methods." In *Proceedings of the 1987 Winter Simulation Conference*, edited by A. Thesen, H. Grant and W. D. Kelton, 409-418. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Tu, J. 2003. "Cross-Validated Multivariate Metamodeling Methods for Physics-based Computer Simulations." In *Proceedings of the IMAC-XXI, A Conference on Structural Dynamics*, The Hyatt Orlando, Kissimee,FL February 3-6, 2003.

Walsh, J. E. 1963. "Use of Linearized Nonlinear Regression for Simulations Involving Monte Carlo." *Operations Research* 11:228-235.

Xie, W., B. L. Nelson and R. R. Barton. 2014. "Statistical Uncertainty Analysis for Stochastic Simulation with Dependent Input Models." In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 674-685. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHY

RUSSELL R. BARTON is professor of supply chain and information systems and professor of industrial engineering at the Pennsylvania State University. He currently serves as senior associate dean for research and faculty in the Smeal College of Business. He received a B.S. degree in electrical engineering from Princeton University and M.S. and Ph.D. degrees in operations research from Cornell University. Before entering academia, he spent twelve years in industry. He is a past president of the INFORMS Simulation Society, and currently serves as the I-Sim representative on the INFORMS Subdivisions Council and chairs the QSR Advisory Board. He is a senior member of IIE and IEEE and a Certified Analytics Professional®. 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.