*Proceedings of the 2014 Winter Simulation Conference*
*A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.*

# STEADY-STATE QUANTILE PARAMETER ESTIMATION: AN EMPIRICAL COMPARISON OF STOCHASTIC KRIGING AND QUANTILE REGRESSION

Jennifer M. Bekki

Engineering and Computing Systems
Arizona State University
Mesa, AZ 85212, USA

Xi Chen

Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24601, USA

Demet Batur

Department of Management
University of Nebraska-Lincoln
Lincoln, NE 68588, USA

## ABSTRACT

The time required to execute simulation models of modern production systems remains high even with todays computing power, particularly when what-if analyses need to be performed to investigate the impact of controllable system input variables on an output performance measure. Compared to mean and variance which are frequently used in practice, quantiles provide a more complete picture of the performance of the underlying system. Nevertheless, quantiles are more difficult to estimate efficiently through stochastic simulation. Stochastic kriging (SK) and quantile regression (QR) are two promising metamodeling tools for addressing this challenge. Both approximate the functional relationship between the quantile parameter of a random output (e.g., cycle time) and multiple input variables (e.g., start rate, unloading times). In this paper, we compare performances of SK and QR on steady-state quantile parameter estimation. Results are presented from simulations of an M/M/1 queue and a more realistic model of a semiconductor manufacturing system.

## 1 INTRODUCTION

Simulation metamodels provide statistically-based approximations of the input-output relationship for a given discrete-event simulation model. The metamodels can be used within a prescribed design space of the input variables to predict parameters (such as mean and quantiles) of the underlying response-variable distribution without having to actually execute the simulation model on the fly. Ideally, a well-built metamodel can provide the fidelity of the full simulation model with the ease of use of a spreadsheet model.

Despite the fact that the *mean* parameter of the response-variable distribution is the most widely considered output measure in the simulation metamodeling literature, *quantiles* provide a much more comprehensive understanding of the response-variable distribution and are arguably more useful to decision makers. We note explicitly that our analysis is on *steady-state* quantile parameters. Assume that the underlying stochastic process is $\{Y_\ell, \ell = 0,1,2,\ldots\}$ and let $F_{\ell,Y_0}(y) = \mathrm{P}(Y_\ell \le y|Y_0)$ be the finite-horizon cumulative distribution function (cdf) of $Y_\ell$ given the initial state $Y_0$. The process is considered to enter *steady state* if $\lim_{\ell\to\infty} F_{\ell,Y_0}(y) = F(y) := \mathrm{P}(Y \le y)$ for some random variable $Y$, regardless of the initial state, $Y_0$. The random variable $Y$ can be interpreted the state of the process when observed far into the future. In our problem context, the steady-state distribution cdf $F(\cdot;\mathbf{x})$ relies on some input variable vector

$\mathbf{x}$. We are interested in estimating the $p$-quantile of the steady-state distribution, $y_p(\mathbf{x})$, as a function of the input vector $\mathbf{x}$, formally defined as $y_p(\mathbf{x}) = \inf\{y : F(y;\mathbf{x}) \geq p\}$ with $p \in (0,1)$.

The purpose of this paper is to review and to compare two simulation metamodeling approaches, Stochastic Kriging (SK) and Quantile Regression (QR), in the context of steady-state quantile parameter estimation, with the twin goals of elucidating their properties and of contrasting them in a concrete context. The remainder of the paper is structured as follows. Section 2 provides a brief review of both SK and QR. Section 3 details results obtained from applying SK and QR to steady-state quantile estimation through simulations of a simple M/M/1 queueing system and of a more realistic model of a semiconductor manufacturing system. In both cases, the output variables are quantiles of the steady-state distribution surrounding total time spent in the underlying system. Section 4 synthesizes findings and gives directions for future research.

## 2 A REVIEW OF STOCHASTIC KRIGING AND QUANTILE REGRESSION

### 2.1 Stochastic Kriging

Standard stochastic kriging (SK) models the simulation response estimate obtained at a design point $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ on the $j$th simulation replication as

$$\mathscr{Y}_j(\mathbf{x}) = \mathsf{Y}(\mathbf{x}) + \varepsilon_j(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \beta + \mathsf{M}(\mathbf{x}) + \varepsilon_j(\mathbf{x}) \ , \tag{1}$$

where $\mathsf{Y}(\mathbf{x})$ represents the unknown true response that we intend to estimate at point $\mathbf{x}_0 \in \Omega$, and the term $\varepsilon_j(\mathbf{x})$ represents the mean zero simulation error realized on the $j$th replication. The simulation errors $\varepsilon_1(\mathbf{x}), \varepsilon_2(\mathbf{x}), \ldots$ are assumed to be IID across replications at a given design point. Notice that the variance of $\varepsilon_j(\mathbf{x})$ may depend on $\mathbf{x}$. The terms $\mathbf{f}(\cdot)$ and $\beta$ are, respectively, a $p \times 1$ vector of known functions of $\mathbf{x}$ and a $p \times 1$ vector of unknown parameters. The term $\mathsf{M}(\cdot)$ represents a mean zero stationary Gaussian random field such that $\mathsf{E}\left[|\mathsf{M}(\mathbf{x})|^2\right] < \infty$ for all $\mathbf{x} \in \Omega$. One can think of $\mathsf{M}(\mathbf{x})$ as being sampled from a space of mappings $\mathbb{R}^d \to \mathbb{R}$, in which functions are assumed to exhibit spatial correlation. Ankenman et al. (2010) refer to the stochastic nature of $\mathsf{M}$ as *extrinsic uncertainty*, in contrast to the *intrinsic uncertainty* represented by $\varepsilon_j(\mathbf{x})$ that is inherent in a stochastic simulation output. Specifically, the spatial covariance function between two points in the random field is typically modeled as

$$\mathsf{Cov}(\mathsf{M}(\mathbf{x}), \mathsf{M}(\mathbf{y})) = \tau^2 \mathscr{R}(\mathbf{x}, \mathbf{y}; \theta) \ , \tag{2}$$

where $\tau^2$ denotes the spatial variance of the random process and $\mathscr{R}(\cdot,\cdot;\theta)$ is the spatial correlation function. The function $\mathscr{R}(\mathbf{x},\mathbf{y};\theta)$ depends on $\mathbf{x}$ and $\mathbf{y}$ only through their difference; and the parameter vector $\theta = (\theta_1, \theta_2, \ldots, \theta_d)^\top$ controls how quickly the spatial correlation between the two points diminishes as they become farther apart in each direction.

An experimental design for stochastic kriging consists of $\{(\mathbf{x}_i, n_i)_{i=1}^k\}$, a set of design points from the design space $\Omega$ to conduct simulation experiments and the corresponding number of replications to apply (or, the number of simulation response estimates to obtain) at each design point. Denote the $k \times 1$ vector of the sample averages of simulation responses by $\bar{\mathscr{Y}} = \left(\bar{\mathscr{Y}}(\mathbf{x}_1), \bar{\mathscr{Y}}(\mathbf{x}_2), \ldots, \bar{\mathscr{Y}}(\mathbf{x}_k)\right)^\top$, and

$$\bar{\mathscr{Y}}(\mathbf{x}_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathscr{Y}_j(\mathbf{x}_i) = \mathsf{Y}(\mathbf{x}_i) + \bar{\varepsilon}(\mathbf{x}_i), \quad i = 1, 2, \ldots, k, \tag{3}$$

in which $\bar{\varepsilon}(\mathbf{x}_i) = {n_i}^{-1} \sum_{j=1}^{n_i} \varepsilon_j(\mathbf{x}_i)$. Standard stochastic kriging builds a linear predictor of the form $\lambda_0 + {\lambda}^\top \bar{\mathscr{Y}}$ to predict the true response $\mathsf{Y}(\mathbf{x}_0)$ at any given point $\mathbf{x}_0$, such that the location dependent weights $\lambda_0$ and $\lambda$ are chosen to minimize the resulting MSE. Ankenman et al. (2010) show that the MSE-optimal predictor of $\mathsf{Y}(\mathbf{x}_0)$ is given by

$$\widehat{\mathsf{Y}}(\mathbf{x}_0) = \mathbf{f}(\mathbf{x}_0)^\top \beta + \Sigma_{\mathsf{M}}(\mathbf{x}_0, \cdot)^\top \Sigma^{-1} \left(\bar{\mathscr{Y}} - \mathbf{F}\beta\right), \tag{4}$$

and its corresponding mean square error follows as

$$\mathrm{MSE}\left(\widehat{\mathsf{Y}}(\mathbf{x}_0)\right) = \Sigma_{\mathsf{M}}(\mathbf{x}_0,\mathbf{x}_0) - \Sigma_{\mathsf{M}}(\mathbf{x}_0,\cdot)^\top \Sigma^{-1} \Sigma_{\mathsf{M}}(\mathbf{x}_0,\cdot), \tag{5}$$

where $\Sigma = \Sigma_{\mathsf{M}} + \Sigma_\varepsilon$, and $\mathbf{F} = \left(\mathbf{f}(\mathbf{x}_1)^\top, \mathbf{f}(\mathbf{x}_2)^\top, \ldots, \mathbf{f}(\mathbf{x}_k)^\top\right)^\top$. To implement SK for prediction, the standard practice is to first substitute $\widehat{\Sigma}_\varepsilon$ into $\Sigma = \Sigma_{\mathsf{M}} + \Sigma_\varepsilon$, with the $i$th diagonal entry of $\widehat{\Sigma}_\varepsilon$ specified by $\widehat{\sigma}_i^2 = (n-1)^{-1}\sum_{j=1}^n \left(\mathscr{Y}_j(\mathbf{x}_i) - \bar{\mathscr{Y}}(\mathbf{x}_i)\right)^2$ for $i = 1,2,\ldots,k$. Prediction then follows (4) and (5) upon obtaining the metamodel parameter estimates through maximizing the log-likelihood function formed under the standard assumption stipulated by SK that $(\mathsf{Y}(\mathbf{x}_0), \bar{\mathscr{Y}}^\top)^\top$ follows a multivariate normal distribution (Ankenman et al. 2010, Chen and Kim 2014).

Chen and Kim (2013) recognize that constructing a SK predictor given in (4) requires two building blocks, namely, point estimates of the desired response measures at all $k$ design points and the corresponding variance estimates. In the case of standard SK, they are respectively given by the vector $\bar{\mathscr{Y}}$ and the diagonal entries of $\widehat{\Sigma}_\varepsilon$ (provided that CRN is not implemented). Chen and Kim (2013) consider alternative ways to create the two building blocks so that better SK predictors of quantile-based performance measures can be achieved. Specifically, they modify (3) to the following form to better characterize the point estimate of the quantile-based performance measure obtained at design point $\mathbf{x}_i$,

$$\tilde{\mathscr{Y}}(\mathbf{x}_i) = \mathsf{Y}(\mathbf{x}_i) + \tilde{\varepsilon}(\mathbf{x}_i) = \mathbf{f}(\mathbf{x}_i)^\top \beta + \mathsf{M}(\mathbf{x}_i) + \tilde{\varepsilon}(\mathbf{x}_i)\ , \quad i = 1,2,\ldots,k, \tag{6}$$

where $\tilde{\mathscr{Y}}(\mathbf{x}_i)$ represents the point estimate of the $p$-quantile obtained by a given estimation method from the $N$ simulated outputs at $\mathbf{x}_i$, $\mathbf{L} := \{Y_j\}_{j=1}^N$; and $\mathsf{Y}(\mathbf{x}_i)$ stands for the true $p$-quantile at $\mathbf{x}_i$. *The key idea is that instead of targeting the mean response surface, by* (6) *the unknown true quantile* $\mathsf{Y}(\mathbf{x}_i)$ *is modeled as a random draw from a Gaussian process, with mean* $\mathbf{f}(\mathbf{x}_i)^\top\beta$ *and its spatial covariance with* $\mathsf{Y}(\mathbf{x}')$ *at another location* $\mathbf{x}'$ *following conveniently from the spatial covariance structure prescribed for the Gaussian random field* $\mathsf{M}(\cdot)$ *in the format of* (2). Furthermore, it is assumed that the simulation error term $\tilde{\varepsilon}(\mathbf{x}_i)$ has mean $\xi(\mathbf{x}_i;\{n_s,n\})$ and intrinsic variance $\tilde{\sigma}^2(\mathbf{x}_i;\{n_s,n\})$, where $n, n_s$ are parameters that satisfy $n_s \cdot n = N$. Notice that the term $\xi(\mathbf{x}_i;\{n_s,n\})$ explicitly accounts for the bias present in the point estimate $\tilde{\mathscr{Y}}(\mathbf{x}_i)$. The bias and intrinsic variance are expected to decrease according to respective decay rates, which are assumed to depend on the specific estimation method implemented. Given that an estimation method is available to provide the point estimates $\{\tilde{\mathscr{Y}}(\mathbf{x}_i)\}_{i=1}^k$ and their corresponding intrinsic variance estimates $\{\widehat{\tilde{\sigma}^2}(\mathbf{x}_i)\}_{i=1}^k$, the SK metamodel construction and prediction can be performed in a similar fashion as for standard SK (Ankenman et al. 2010, Chen and Kim 2014).

Through some examples Chen and Kim (2013) study the performances of SK for quantile estimation with different estimation methods implemented, namely, batching, sectioning, sectioning-batching, jackknifing variance estimation and jackknifing bias-correction methods. They recommend to use *sectioning* and *section-batching* with SK for quantile estimation. The interested reader is referred to Chen and Kim (2013) and references therein for details. In this paper, we focus on applying SK with sectioning implemented for quantile estimation. Specifically, the method *sectioning* (Asmussen and Glynn 2007, Nakayama 2012) suggests to divide the entire sample $\mathbf{L}$ generated at a design point into $n$ sections each of size $n_s$ (assuming $N = n \cdot n_s$), and construct the $p$-quantile point and variance estimates based on the $n$ quantile estimates as follows. Denoting the $j$th section of simulation outputs by $\mathbf{L}^{(j)} := \{Y_{(j-1)n_s+h}\}_{h=1}^{n_s}$ for $j = 1,2,\ldots,n$, the $p$-quantile point estimate and its corresponding variance estimate are given by

$$\widehat{v}_p^{\,\mathrm{sect}} = \Phi(\mathbf{L}), \quad \widehat{\sigma}^2_{\mathrm{sect}} = \frac{1}{n(n-1)}\sum_{j=1}^n \left(\Phi\left(\mathbf{L}^{(j)}\right) - \widehat{v}_p^{\,\mathrm{sect}}\right)^2, \tag{7}$$

where the operator $\Phi(\cdot)$ maps a sample of size $m$ to its $\lceil pm \rceil$th order statistic, and the ceiling function $\lceil a \rceil$ gives the smallest integer not less than $a$. Hence $\Phi(\mathbf{L})$ and $\Phi\left(\mathbf{L}^{(j)}\right)$ represent the sample $p$-quantiles obtained from the entire sample $\mathbf{L}$ and the $j$th section $\mathbf{L}^{(j)}$, respectively.

### 2.2 Quantile Regression

Quantile Regression (QR) models the relationship between a set of input variables and specific quantiles of the response variable (Koenker (2005)). It is defined as the solution to the problem of minimizing a weighted sum of absolute residuals. The $p$-quantile in a sample $\{Y_i\}_{i=1}^N$ can be computed with

$$\min_\xi \sum_{i=1}^{N} \Big( pI(Y_i > \xi) + (1-p)I(Y_i < \xi) \Big) |Y_i - \xi|, \tag{8}$$

where $I(\cdot)$ denotes the indicator function, which takes a value of one if the event is true and zero otherwise. In QR, the $\xi$ in (8) is replaced by $g(\mathbf{x}_i) = \mathbf{f}(\mathbf{x}_i)^\top \beta$, where $\mathbf{f}$ is a vector of known functions of $\mathbf{x}$ and $\beta$ is a vector of unknown coefficients. The resulting minimization problem can be solved very efficiently by linear programming methods, and the resulting regression fit, $\widehat{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \widehat{\beta}$, provides an estimate of the $p$-quantile of the response-variable distribution given $\mathbf{x}$.

In the proposed QR-based metamodeling procedure, a polynomial regression function is utilized. Polynomial regression functions are preferred for situations where little is known about the response surface and are considered to be linear regression models because they are linear in their unknown coefficients and polynomial in the input variables. The procedure we present utilizes QR with L1-regularization, or the Lasso penalty (Tibshirani (1996)), which imposes a bound on the sum of the absolute value of regression model coefficients (excluding the intercept coefficient). The details of the QR procedure are as follows.

**Initialization**: Determine the vector of input variables, $\mathbf{x}$, and the output performance variable, $Y$. Set $p$, $0 < p < 1$, to be the desired quantile of the steady-state distribution of $Y$. Determine the set of design points $\mathbf{x}_i$, $i = 1, 2, \ldots, k$, at which to observe the response variable $Y$. Initialize the order of the polynomial, $v$, to 2 and the Lasso parameter, $\lambda$, to 0.01. Initialize the value of the average estimation error, $E_{v,\text{old}}$, to a large number such as 1000.

**Simulation**: Run the simulation model at each of the $k$ design points to generate $N$ IID observations. Let $\{Y_{ij}, i = 1, 2, \ldots, k, j = 1, 2, \ldots, N\}$ be the output observations. Let $Y_{i[1]}, Y_{i[2]}, \ldots, Y_{i[N]}$ be the corresponding order statistics in ascending order. At each design point $\mathbf{x}_i$, $i = 1, 2, \ldots, k$, compute the order statistic-based estimate of the $p$-quantile (OS) as shown in (9), where $\lfloor z \rfloor$ denotes the integral part of the real number $z$,

$$\widehat{y}_p(\mathbf{x}_i) = Y_{i[\lfloor Np+1 \rfloor]} \ . \tag{9}$$

**Quantile Regression Fit**: Using the QR method with Lasso and the current values of $v$ and $\lambda$, fit a $v$th order polynomial metamodel function, $\widehat{g}(\mathbf{x})$, where $\widehat{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^{\mathrm{T}} \widehat{\beta}$. The terms of the $\mathbf{f}(\mathbf{x})$ vector are the terms of a full $v$th order polynomial except the single $v$th power terms such as $x_w^v$, where $x_w$ are the input variables,

$$\{x_w : \forall w\} \cup \{ \prod_{\forall w} x_w^{r_w} : r_w > 0, \sum_{\forall w} r_w \leq v \}. \tag{10}$$

**Error Calculation**: At each design point $\mathbf{x}_i$, $i = 1, 2, \ldots, k$, compute the QR estimate of $p$-quantile from $\widehat{g}(\mathbf{x}_i)$. Compute $E_v$, the average error of the QR estimates with respect to the OS quantile estimates, $\widehat{y}_p(\mathbf{x}_i)$ as follows

$$E_v = \frac{1}{k} \sum_{i=1}^{k} \frac{|\widehat{g}(\mathbf{x}_i) - \widehat{y}_p(\mathbf{x}_i)|}{\widehat{y}_p(\mathbf{x}_i)}.$$

**Polynomial Order Fixation**: If the polynomial order, $v$, is not fixed yet, do the following: If $E_v < E_{v,\text{old}}$, increase $v$ by one, set $E_{v,old} = E_v$, and go back to the Quantile Regression Fit step. Else if $E_v \geq E_{v,\text{old}}$, fix the polynomial order at $v-1$, set $\widehat{g}(\mathbf{x})$ and $E_v$ to their previous values, and initialize $E_{v,old}$ to 1000.

**Lasso Parameter Fixation**: If the Lasso parameter, $\lambda$, is not fixed yet, do the following: If $E_v < E_{v,\text{old}}$, divide $\lambda$ by 10, set $E_{v,old} = E_v$, and go back to the Quantile Regression Fit step. Else if $E_v \geq E_{v,\text{old}}$, fix the Lasso parameter at $\lambda/10$.

**Stopping Condition**: Stop and return the function $\widehat{g}(\mathbf{x})$ for the $p$-quantile fit using the current $v$ and $\lambda$ values. □

In the Initialization step of the QR procedure, the input variables and the quantile of interest are determined. Similar to SK, given a set of $k$ design points, simulation runs are performed at each one of them and the random outputs are collected. Additionally, the order of polynomial regression function, $v$; the Lasso parameter, $\lambda$; and the average estimation error, $E_{v,\text{old}}$, are initialized at 2, 0.01, and 1000 respectively. In the following steps of the procedure, the $v$ and $\lambda$ values are changed iteratively until no further reduction in the average error, $E_v$, is achieved. In the Simulation step, the simulation model is run at each design point to generate $N$ IID output observations. At each design point, order statistic-based (OS) quantile estimates are computed. In the Quantile Regression Fit step, a $v$th order polynomial metamodel function, $\widehat{g}(\mathbf{x})$, is fit using QR with Lasso based on the current $\lambda$ value. In the Error Calculation step, $\widehat{g}(\mathbf{x})$ is used to obtain the QR estimate of the $p$-quantile at each design point, and the average error, $E_v$, of the QR quantile estimates is computed with respect to the OS quantile estimates at the $k$ design points. Finally, in the Polynomial Order Fixation and Lasso Parameter Fixation steps, the values for the polynomial order, $v$, and Lasso parameter, $\lambda$, that result in the least average estimation error are selected. Once the best parameter values are determined, the procedure stops, returning the $\widehat{g}(\mathbf{x})$ function representing QR function of polynomial order, $v$, and Lasso parameter, $\lambda$. Notice that once the parameters $v$ and $\lambda$ for a particular system have been determined through execution of the QR procedure for the $p$-quantile, the same set of parameters can be used to estimate other quantiles for simulating the same system without having to execute the procedure again.

## 3 EXPERIMENTS

We present next the results of applying the SK and QR metamodeling approaches to output data from discrete-event simulation models of a simple M/M/1 queueing system and a more realistic production system. All simulation experiments are conducted using Arena simulation software. The QR method is applied using the *quantreg* packing within R, and the SK method is implemented using MATLAB.

### 3.1 Experiments on an M/M/1 Queueing System

Two major issues that steady-state simulation output analysis has to deal with are the initial transient bias and dependence in the output data. By simulating an M/M/1 queue, we investigate the impact of the two aspects on the predictive accuracy achieved by SK and QR via four separate experiments. To this end, we control the following experimental conditions: system initialization condition, truncation of the transient period, and the degree of dependence in the data used for metamodel construction and prediction through spacing observations. In addition, the effects of the number of design points relative to the size of the design space and the number of observations collected at each point $N$ for metamodel construction on the performances of SK and QR are also considered. For all experiments, we are interested in estimating quantiles of the distribution surrounding the steady-state sojourn time (i.e., total time in the system). The service rate is set to 1 part per time unit, and the single input variable is system utilization, controlled by the arrival rate. Notice that in each experiment we simulate one single long sample path rather than multiple shorter sample paths.

The experimental conditions for all four experiments are summarized in Table 1. In each experiment, simulations are run at each of $k$ design points equally spaced within the respective design space given in Table 1. We control the random seeds such that a fair comparison can be made between results obtained under different experimental conditions for a given experiment. In addition, different random seeds are applied at distinct design points within a given experiment to avoid generating CRN-induced correlation, which has been linked to degradation of predictive accuracy of the metamodeling techniques in the literature (e.g., Chen et al. (2012)). The system is initialized with either (1) zero or (2) the steady-state mean number of entities present in the queue. Truncation of the initial transient data is performed for all experimental

conditions other than 1.3 and 1.4. The initialization phase is determined by plotting the cumulative average sojourn time of the generated sample path (Banks et al. 2010). For initializations in Condition (1), the first 250,000 consecutive observations are truncated, and for (2) the first 120,000 time observations are deleted. For a given experiment, a total of $N$ observed sojourn times are obtained. If spacing in the output sequence is applied, then the $N$ observations are pseudo-independent, achieved by employing a lag length of 4100 observations between collected data points. This lag length is identified through studying a plot of the output sequence giving auto-correlation vs. lag length.

Table 1: A summary of the conditions used for simulating the M/M/1 queue is given. Identical experimental conditions are denoted by $*$, and through controlling the random seeds, the obtained random outputs are the same.

| Index | Design space | # design points | Initialize at zero | Initialize at mean | Truncation | Spacing | # obs. collected |
|---|---|---|---|---|---|---|---|
| 1.1 | $[0.65, 0.95]$ | 10 | $\surd$ | | Yes | Yes | $N$ |
| $*$1.2 | $[0.65, 0.95]$ | 10 | | $\surd$ | Yes | Yes | $N$ |
| 1.3 | $[0.65, 0.95]$ | 10 | $\surd$ | | No | Yes | $N$ |
| 1.4 | $[0.65, 0.95]$ | 10 | | $\surd$ | No | Yes | $N$ |
| 2.1 | $[0.4, 0.95]$ | 10 | | $\surd$ | Yes | Yes | $N$ |
| $*$2.2 | $[0.65, 0.95]$ | 10 | | $\surd$ | Yes | Yes | $N$ |
| $*$3.1 | $[0.65, 0.95]$ | 10 | | $\surd$ | Yes | Yes | $N$ |
| 3.2 | $[0.65, 0.95]$ | 10 | | $\surd$ | Yes | No | $4100(N-1)+N$ |
| 3.3 | $[0.65, 0.95]$ | 10 | | $\surd$ | Yes | No | $N$ |
| 4.1 | $[0.65, 0.95]$ | 9 | | $\surd$ | Yes | Yes | $N$ |
| 4.2 | $[0.65, 0.95]$ | 5 | | $\surd$ | Yes | Yes | $N$ |

In each experiment, the SK and QR metamodels are built to predict the 0.6, 0.8, 0.9, and 0.95 quantiles of steady-state sojourn time over the corresponding design space. We note that all metamodels built by SK use sectioning with $n = 25$ sections. All metamodels fit by QR are fifth-order polynomials with $\lambda = 10^{-13}$, except for Condition 4.2 where $\lambda = 10^{-4}$. The prediction-point set consists of 1000 equally spaced points in [0.4, 0.95]. Given the arrival rate $x \in [0.4, 0.95]$, the close-form expression of the $p$-quantile of steady-state sojourn time distribution is given by $y_p(x) = -\ln(1-p)/(1-x)$ for $p \in (0,1)$. The predictive accuracy is evaluated through the mean absolute percentage error (MAPE). Notice that the entire prediction-point set is not relevant for all experiments. Only those points of the original 1000 that are contained within the corresponding design space for a given experiment are utilized in calculating the MAPE.

### 3.1.1 Results

**Experiments 1** tests the impact of the initial transient bias in the output sequence on the predictive performances of SK and QR. A summary of the MAPEs resulting from the quantile estimates obtained by SK and QR using $N = 500$ simulation observations are shown in Table 2. The results show that the MAPEs achieved by QR are typically smaller than those obtained by SK, but the differences are not substantial. We observe that truncation does seem to help reduce the MAPEs in the case of initiating the system empty (Conditions 1.1 vs. 1.3), but does not help as much in the case of initiating the system at the steady-state mean number (Conditions 1.2 vs. 1.4). It is also interesting to note that the MAPEs obtained by SK and QR for estimating 0.8-quantiles are smaller than those for estimating the 0.6-quantiles given the random outputs collected along a single long sample path. Under the same experimental conditions, we take a closer look at the predictive performances of SK and QR as a function of $N$, i.e., by looking at $N = 250, 500$, and 1000. For the sake of brevity, we mention without showing details that the resulting MAPEs obtained by both QR and SK increase moderately as $N$ decreases. We also note that though the overall effect of

truncation is not obvious with $N = 1000$, as $N$ decreases to 500 and further to 250, the truncation effect seems to become more evident.

Table 2: A summary of MAPEs obtained by SK and QR for Experiment 1 with $N = 500$.

| Quantile | 1.1 | | 1.2 | | 1.3 | | 1.4 | |
|---|---|---|---|---|---|---|---|---|
| | SK | QR | SK | QR | SK | QR | SK | QR |
| 0.6 | 3.01% | 2.43% | 4.51% | 3.84% | 3.77% | 3.42% | 4.20% | 3.58% |
| 0.8 | 2.84% | 2.18% | 1.85% | 2.09% | 3.08% | 2.74% | 3.71% | 3.23% |
| 0.9 | 4.31% | 3.12% | 2.11% | 2.44% | 4.73% | 4.31% | 2.37% | 2.72% |
| 0.95 | 3.39% | 3.43% | 2.14% | 2.71% | 3.17% | 2.83% | 1.26% | 1.56% |
| Average | 3.39% | 2.79% | 2.65% | 2.77% | 3.69% | 3.32% | 2.88% | 2.77% |

**Experiment 2** compares the performances of SK and QR in predicting steady-state quantiles over a relatively large design space (i.e., $[0.4, 0.95]$) versus a small one (i.e., $[0.65, 0.95]$). For the sake of brevity, we omit the results for individual quantile estimation and summarize the averaged MAPEs obtained with varying numbers of observations collected, $N$, in columns 3 and 4 of Table 3. With regard to Conditions 2.1 and 2.2, we find that the MAPEs obtained by QR are typically smaller as compared to SK when predicting over $[0.4, 0.95]$, and the advantage of QR over SK becomes more evident as $N$ further decreases. SK, on the other hand, seems to outperform QR when predicting over the smaller region $[0.65, 0.95]$, especially with smaller $N$. With the same number of design points ($k = 10$) evenly spread out in respective prediction regions of interest, SK obtains smaller MAPEs when predicting over the smaller region $[0.65, 0.95]$ as compared to predicting over the larger region $[0.4, 0.95]$, which is as we expected. In strong contrast, QR does better in predicting over the larger region than the smaller region, which is a bit surprising.

Table 3: A summary of the averaged MAPEs obtained for estimating $0.6, 0.8, 0.9$ and 0.95-quantiles for Experiments 2 to 4 with $N = 1000, 500$ and 250 is given. Identical experimental conditions are denoted by $*$.

| $N$ | | 2.1 | 2.2* | 3.1* | 3.2 | 3.3 | 4.1 | 4.2 |
|---|---|---|---|---|---|---|---|---|
| 1000 | SK | 3.61% | 3.21% | 3.21% | 0.64% | 17.91% | 3.33% | 12.25% |
| | QR | 2.71% | 2.59% | 2.59% | NA | 16.40% | 2.80% | 3.62% |
| 500 | SK | 4.75% | 2.65% | 2.65% | 0.69% | 22.13% | 3.02% | 10.54% |
| | QR | 2.69% | 2.77% | 2.77% | NA | 21.10% | 3.91% | 4.61% |
| 250 | SK | 6.40% | 4.73% | 4.73% | 1.06% | 37.96% | 4.52% | 11.15% |
| | QR | 3.73% | 4.89% | 4.89% | NA | 30.19% | 5.40% | 4.98% |

**Experiment 3** studies the impact of correlations in the collected observations on the predictive performances of SK and QR. Notice that a total number of $(4100 \times (N-1)) + N$ consecutive observations are generated (beyond the truncation point) for Experiment 3. Conditions 3.1 to 3.3 are different in their ways of selecting observations from this output sequence for metamodel construction and prediction. Under Condition 3.1, spacing is applied and only $N$ pseudo-IID observations are retained. Under Condition 3.2, spacing is not applied and the entire sequence of consecutive observations is retained. Under Condition 3.3, spacing is not applied but we only retain the first $N$ consecutive observations from the sequence. As such, under Conditions 3.1 and 3.2 an equivalent amount of simulation effort is expended in conducting the simulation, while under Conditions 3.1 and 3.3 the same number of observations are retained for metamodel construction and prediction. The averaged MAPEs obtained for estimating individual quantiles with varying numbers of simulation outputs $N$ are summarized in columns 5, 6, and 7 of Table 3. Notice that the prediction results by QR under Condition 3.2 are unavailable (denoted by 'NA'). While the QR approach is theoretically capable of delivering quantile estimates, practically the R package used was not able to

process the entire sequence of consecutive observations generated. SK, however, can be applied without difficulty and we observe that much smaller MAPEs can be obtained by using the entire sequence than those obtained under Condition 3.1 despite the possibly stronger influence of correlations in the consecutive output sequence. Lastly, for both SK and QR, much larger MAPEs are obtained under Condition 3.3 as compared to Condition 3.1, indicating the degradation of performance when using correlated random observations to fit SK and QR metamodels for quantile estimation; this adverse impact becomes severer as $N$ decreases.

**Experiment 4** investigates the impact of the number of design points used on the predictive performances of SK and QR. We summarize the MAPEs for estimating individual quantiles with varying numbers of simulation outputs $N$ in the last two columns of Table 3. We find that with $N = 500$ and 250, SK delivers smaller MAPEs than QR does when sufficient number of design points are used. It is revealing to compare the results for Condition 4.2 with those for 4.1, since although the predictive performances of both SK and QR deteriorate as the number of design points decreases, the impact on SK appears much more significant. Reducing the number of design points from 9 down to 5 almost triples its resulting MAPEs. This observation confirms the rule of thumb given by earlier studies (e.g., Loeppkya et al. (2009)) that the number of design points for an effective computer experiment should be about 10 times the input dimension; for Condition 4.2, 5 design points are simply not sufficient for SK to achieve adequate prediction accuracy. We also draw some attention to the fact that the value of the tuning parameter, $\lambda = 10^{-4}$, used by QR in Condition 4.2, is different from $\lambda = 10^{-13}$, which is used in Conditions 1.1 to 4.1. Although specifying an appropriate value for $\lambda$ is not a trivial task, the results here indicate that it may be helpful in providing the flexibility required for handling a sparse grid design such as the one given in Condition 4.2.

### 3.2 Experiments on a Production System

Semiconductor manufacturing remains one of the most complicated production environments, and discrete-event simulation models are regularly utilized within this industry to evaluate *what-if* scenarios that consider alternate or future operating conditions. Simulation models of such production systems, however, typically take a very long time to execute, making the use of metamodels particularly attractive. Simulation metamodeling has already been applied to study production systems; recent examples include Kesen et al. (2009), MacDonald and Gunn (2011), and Yang (2010). In this subsection, we utilize the Minifab model as a vehicle to evaluate the performances of SK and QR in predicting quantile parameter of the steady-state cycle time (CT) distribution in more realistic production environments.

The Minifab model is a simplified model of a semiconductor manufacturing facility designed to capture in a simple format the key characteristics that make the modeling and scheduling of semiconductor manufacturing processes particularly difficult: re-entrant flow, batching, setups, preventative maintenance, emergency maintenance, and multiple part types. While still capturing the most important system characteristics, the execution time for the Minifab model is much less than that of a full semiconductor wafer fabrication facility, making it an attractive and common choice for researchers interested in the application area of semiconductor manufacturing (Mönch et al. 2013).

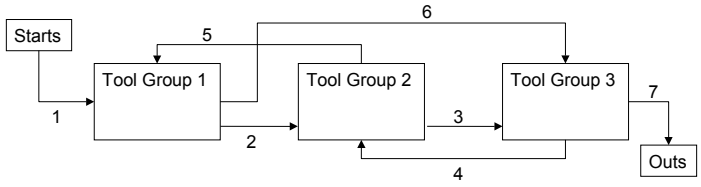



Figure 1: The product flow through the Minifab model is provided.

The Minifab model includes five machines arranged in three tool groups. Two part types, Products X and Y, travel through the system, visiting each tool group twice in the order as shown in Figure 1. The

inter-arrival times of parts of each product type follows an exponential distribution. Tool Group 1 is made up of two parallel processing machines that employ a batch processing system (similar to a diffusion oven in a semiconductor manufacturing system). Tool Group 2 is made up of two machines operating in parallel (e.g., a photo-lithography stepper), while Tool Group 3 is made up of a single machine characterized by sequence-dependent setups (e.g., an ion implanter). Batches processed during Step 1 can include both Products X and Y, but lots waiting to be processed for Step 5 cannot include different product types. Additionally, lots waiting for Steps 1 and 5 at Tool Group 1 can never be batched together. The processing times at each station follow a normal distribution, with a variability of 5% of the mean processing time. Operators serve as resources in the model, and there is one operator for each tool group. Operators are utilized for loading, unloading, and machine setup activities. Each time a new lot (or batch for Tool Group 1) begins or completes processing, the machine needs to be loaded and unloaded. All tool groups require preventative maintenance, and the machine in Tool Group 3 also requires emergency maintenance. Finally, First-In-First-Out (FIFO) dispatching policies are employed at all workstations. Specific numeric attributes of the Minifab model are given in Table 4.

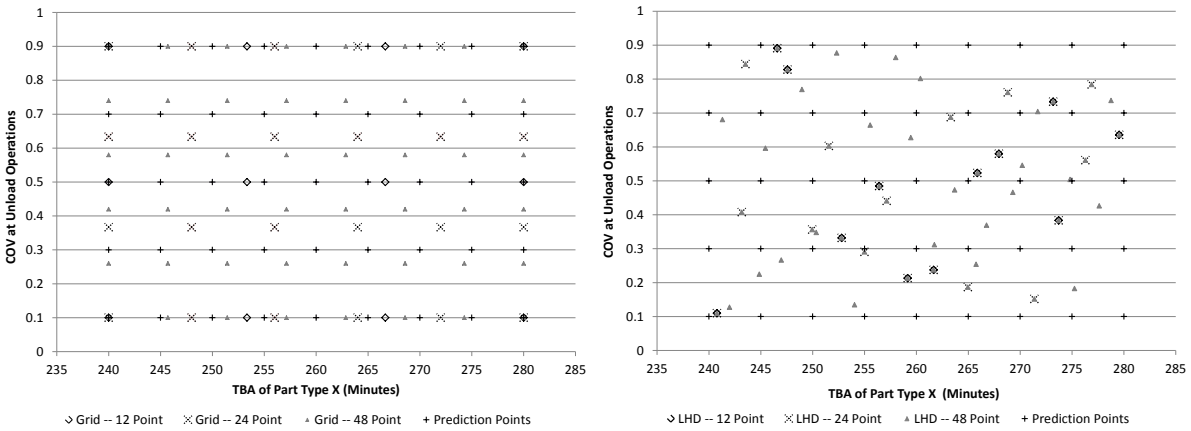
Table 4: The batch sizes, the distributions of processing times, loading time, and unloading times are specified for each of the three tool groups in the Minifab model.

| Process Step | Tool Group | Processing Time (min) | Batch Size (lots) | Load Time (min) | Unload Time (min) |
|---|---|---|---|---|---|
| 1 | 1 | $N(225, 11.25)$ | 3 | $N(20, 2)$ | $N(40, 4)$ |
| 2 | 2 | $N(30, 1.5)$ | 1 | $N(15, 1.5)$ | $N(15, 1.5)$ |
| 3 | 3 | $N(55, 2.75)$ | 1 | $N(10, 1)$ | $N(10, 1)$ |
| 4 | 2 | $N(50, 2.5)$ | 1 | $N(15, 1.5)$ | $N(15, 1.5)$ |
| 5 | 1 | $N(255, 12.75)$ | 3 | $N(20, 2)$ | $N(40, 4)$ |
| 6 | 3 | $N(10, 0.5)$ | 1 | $N(10, 1)$ | $N(10, 1)$ |

Quantiles of the steady-state cycle time (CT) distribution for actual production facilities are typically driven by more than one input variable. In this example, two input variables are utilized: start rate (which drives throughput / system utilization) and the coefficient of variation (COV) at all unload operations. The start rate is controlled simply by changing inter-arrival times of Part X to the system, and the COV at the unload operations can be manipulated easily by changing the standard deviations given in the final column of Table 4. Although not considered here, we note that other variables such as inter-failure times, repair times, setup times, and loading times, etc., could also be used as input variables.

To determine the experimental points at which to execute the Minifab model, a two-dimensional experimental region $\Omega = [240, 280] \times [0.1, 0.9]$ is considered in which the first dimension corresponds to the start rate and the second dimension corresponds to the COV at all unload operations. We note that such an experimental region implies that the system utilization roughly ranges from 0.85 to 0.95. For metamodel construction and prediction, we run simulations at $k$ distinct design points selected from $\Omega$ following two types of designs, namely, grid designs (GD$k$) and Latin Hypercube designs (LHD$k$). To evaluate the effects of number of design points, we choose the values of $k$ in $\{12, 24, 48\}$. To maintain the space-filling property of LHD with only a few points scattered in the two-dimensional design space, we adopt a three-layer nested LHD (Qian (2009)) with 48 points to construct LHD12, LHD24 and LHD48. That is, the first layer is itself a LHD with 12 points, the second layer is a LHD with 24 points, and the full design is a LHD with 48 points. The three increasingly denser grid designs, Grid12, Grid24 and Grid48, on the other hand, are not nested. For a given design, 1000 pseudo-IID observations of CT are collected through simulations at each design point. We note that appropriate truncation and spacing are applied to the output sequence to generate these pseudo-IID observations. SK and QR are then applied to estimate the 0.5, 0.8, 0.9, and 0.95 steady-state quantiles of the steady-state CT distribution.

The predictive accuracy of SK and QR is evaluated based on the MAPE calculated based on the quantile estimates given by SK and QR and approximated true quantiles obtained using order-statistics based estimates from very intensive simulations at each of 45 prediction points. Al metamodels built by SK use sectioning with $n = 25$ sections. All metamodels fit by QR use fourth-order polynomials; for the GD48 and GD24 designs $\lambda = 10^{-13}$, while for the GD12 design, $\lambda = 0.01$. Figures 2(a) and (b) give the design points utilized for both the grid designs and the LHDs along with the locations of the prediction points.




(a) Three-level GD

(b) Three-layer nested LHD

Figure 2: Figures (a) and (b) show the prediction-point locations, the design-point locations in the grid designs and in the three-layer nested LHD.

### 3.2.1 Results

The MAPEs obtained from the Minifab experimentation using the grid designs and LHDs are presented in Tables 5 and 6, respectively. For the grid designs, it is interesting to notice that in contrast to the results obtained for the M/M/1 example where QR leads to lower MAPEs than SK does in most cases, here the performances of the two are pretty close; it is only when the design is densest (GD48), the advantage of QR becomes noticeable. As to using the LHDs, it is surprising to see that SK outperforms QR with respect to estimating almost all the quantiles. A closer look reveals that this is largely due to the notable degradation in the performance of QR from using grid designs to LHDs. In contrast, the resulting degradation from the change of design type is much less pronounced for SK. As expected, regardless of design types used, increasing the number of design points leads to better predictive accuracy of SK and QR. While the results demonstrate the potential of SK and QR serving as effective metamodeling approaches for approximating quantile surfaces across a higher-dimensional design space, it is noteworthy that there is clearly a difference in terms of appropriate choices of experimental designs for SK and QR to achieve adequate predictive performances.

## 4 CONCLUSIONS

This paper demonstrates and compares the performances of two metamodeling approaches, SK and QR, in the context of steady-state quantile parameter estimation through simulations of a simple M/M/1 queueing system and a more realistic production system model. Both SK and QR are found to be effective in metamodeling quantile response surfaces over a higher-dimensional design space. It is observed that typically QR handles sparse grid designs better than SK does, while SK works better with space-filling

Table 5: A summary of the predictive accuracy, measured by MAPE, achieved by SK and QR based on GD12, GD24 and GD48 is given.

| Quantile | GD48 | | GD24 | | GD12 | |
|---|---|---|---|---|---|---|
| | SK | QR | SK | QR | SK | QR |
| 0.5 | 2.37% | 2.01% | 2.88% | 3.65% | 4.65% | 15.23% |
| 0.8 | 2.40% | 2.79% | 3.74% | 4.67% | 7.78% | 6.13% |
| 0.9 | 4.11% | 3.03% | 4.12% | 6.29% | 7.75% | 7.19% |
| 0.95 | 4.58% | 3.33% | 6.32% | 7.13% | 11.07% | 7.59% |
| Average | 3.36% | 2.79% | 4.26% | 5.43% | 7.81% | 9.04% |

Table 6: A summary of the predictive accuracy, measured by MAPE, achieved by SK and QR based on the three-layer LHD with 48 points is given.

| Quantile | LHD48 | | LHD24 | | LHD12 | |
|---|---|---|---|---|---|---|
| | SK | QR | SK | QR | SK | QR |
| 0.5 | 2.51% | 3.61% | 3.82% | 7.51% | 9.44% | 17.92% |
| 0.8 | 3.13% | 3.38% | 3.98% | 9.47% | 7.57% | 22.19% |
| 0.9 | 4.28% | 4.26% | 5.25% | 14.20% | 9.67% | 24.49% |
| 0.95 | 4.01% | 4.08% | 5.56% | 11.00% | 7.41% | 30.56% |
| Average | 3.48% | 3.84% | 4.65% | 10.54% | 8.52% | 23.79% |

designs such as LHDs than QR does. For both approaches, there remains much more to explore regarding appropriate experimental designs for them to achieve adequate predictive accuracy.

Our findings on metamodeling for steady-state simulation via SK and QR are briefly summarized as follows. Truncation of the initial transient period seems to have a greater impact than the initialization condition does, but implementing both helps improve performances of SK and QR. The impact of dependence in the output sequence on the performances of SK and QR is profound, particularly when only a smaller number of observations are used for metamodel construction and prediction. On the other hand, using all the output data without spacing the observations is a viable choice, but it works only if a very large number of observations are generated so that the impact of dependence becomes relatively minor. Results for Experiment 3.2 does show some limitations of the current QR approach; we anticipate this is something that could be addressed through future research.

Lastly, we note that for all experiments conducted a single long simulation sample path is simulated rather than multiple shorter ones. Therefore, our findings are by no means conclusive. In addition to studying metamodeling based on multiple simulated sample paths, future work will include the investigation of efficient experimental designs for both SK and QR to account for the sampling variability across the design space of interest with a given simulation budget.

## REFERENCES

Ankenman, B. E., B. L. Nelson, and J. Staum. 2010. "Stochastic kriging for simulation metamodeling". *Operations Research* 58:371–382.

Asmussen, S., and P. W. Glynn. 2007. *Stochastic Simulation*. New York: Springer.

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

Chen, X., B. E. Ankenman, and B. L. Nelson. 2012. "The effects of common random numbers on stochastic kriging metamodels". *ACM Transactions on Modeling and Computer Simulation* 22:7/1–7/20.

Chen, X., and K.-K. Kim. 2013. "Building metamodels for quantile-based measures using 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, Inc.

Chen, X., and K.-K. Kim. 2014. "Stochastic kriging with biased sample estimates". *ACM Transactions on Modeling and Computer Simulation* 24:8/1–8/23.

Kesen, S., M. D. Toksari, Z. Güngor, and E. Güner. 2009. "Analyzing the behaviors of virtual cells (VCs) and traditional manufacturing systems: ant colony optimization (ACO)-based metamodels". *Computers & Operations Research* 36:2275–2285.

Koenker, R. 2005. *Quantile Regression*. New York: Cambridge University Press.

Loeppkya, J. L., J. Sacksb, and W. J. Welch. 2009. "Choosing the sample size of a computer experiment: a practical guide". *Technometrics* 51:366–376.

MacDonald, C., and E. Gunn. 2011. "A framework for analysis and production authorization card-controlled production systems". *Production and Operations Management* 20 (6):937–947.

Mönch, L., J. Fowler, and S. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. New York: Springer.

Nakayama, M. K. 2012. "Using sectioning to construct confidence intervals for quantiles when applying importance sampling". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Qian, P. 2009. "Nested latin hypercube designs". *Biometrika* 96 (4):957–970.

Tibshirani, R. 1996. "Regression shrinkage and selection via the lasso". *Journal of the Royal Statistical Society B* 58 (1):267–288.

Yang, F. 2010. "Neural network metamodeling for cycle time-throughput profiles in manufacturing". *European Journal of Operational Research* 205:172–185.

## AUTHOR BIOGRAPHIES

**JENNIFER M. BEKKI** is an Assistant Professor in the Department of Engineering & Computing Systems at Arizona State University. Her research interests include simulation methodology, the modeling and analysis of manufacturing systems, and educational data mining. Her email address is jennifer.bekki@asu.edu.

**XI CHEN** is an Assistant Professor in the Grado Department of Industrial and Systems Engineering at Virginia Tech. Her research interests include stochastic modeling and simulation, applied probability and statistics, computer experiment design and analysis, and simulation optimization. Her email address is xchen6@vt.edu.

**DEMET BATUR** is an Assistant Professor in the Department of Management at the University of Nebraska-Lincoln. Her research interests are in simulation methodology, stochastic decision analysis, and supply chain and manufacturing systems. Her email address is dbatur@unl.edu and her web page is http://cba.unl.edu/people/dbatur/.