Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

STOCHASTIC CO-KRIGING FOR STEADY-STATE SIMULATION METAMODELING

Xi Chen

Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061, USA

Sahar Hemmati
Feng Yang

Industrial and Management Systems Engineering
West Virginia University
Morgantown, WV 26506, USA

ABSTRACT

In this paper we present the stochastic co-kriging methodology (SCK) for approximating a steady-state mean response surface based on outputs from both long and short simulation replications performed at selected design points. We provide details on how to construct an SCK metamodel, perform parameter estimation, and make prediction via SCK. We demonstrate numerically that SCK holds the promise of providing more accurate prediction results at no additional computational effort by only externally adjusting the simulation runlength and number of independent replications of simulations through the experimental design of the simulation study.

1 INTRODUCTION

In deterministic computer experiments, a computer code can often be run at different levels of complexity/fidelity and a hierarchy of levels of code can be obtained. The higher the fidelity and hence the computational cost, the more accurate output data can be obtained. Methods based on the co-kriging methodology (Cressie 2015) for predicting the output of a high-fidelity computer code by combining data generated to varying levels of fidelity have flourished over the last two decades. For instance, Kennedy and O'Hagan (2000) first propose to build a metamodel for multi-level computer codes by using an autoregressive model structure. Forrester et al. (2007) provide details on estimation of the model parameters and further investigate the use of co-kriging for multi-fidelity optimization based on the efficient global optimization algorithm (Jones et al. 1998). Qian and Wu (2008) propose a Bayesian hierarchical modeling approach for combining low-accuracy and high-accuracy experiments. More recently, Gratiet and Cannamela (2015) propose sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes.

In the context of stochastic simulation, steady-state simulations are often employed for studying long-run system behavior, and they play a significant role in system design and risk assessment. Long-run performance of stochastic systems such as telecommunication networks is often evaluated by steady-state mean and quantiles of the system's response times (Jeong et al. 2005). Therefore, estimation of steady-state parameters of complex stochastic systems is of great interest to simulation researchers and practitioners.

There exists a plethora of work on point or interval estimation of mean performance measure implied by a steady-state simulation. Various data collection and analysis methods have been proposed to overcome the two challenges arising from output analysis of a steady-state simulation, namely, the initial bias in the sample mean as a point estimator caused by the initial conditions and the difficulty in estimating the variance of the sample mean due to correlations in the sequence of outputs from within a single replication. Existing variance estimation methods include those based on independent replications (IR), batch means (BM), overlapping batch means (OBM), uncorrelated sampling, regenerative cycles, spectral analysis, autoregressive representation and standardized time series, etc.; see Pawlikowski (1990) for a survey on various methods proposed for steady-state queueing simulations by early 1990's. More recently,

Argon et al. (2013) propose the replicated batch means approach (RBM), known as a compromise method between IR and BM (Alexopoulos and Goldsman 2004).

Assuming a simulation budget constraint given in terms of simulation clock time or the number of discretely-indexed observations, a decision must be made before the simulations are run as to the number of independent, identically initialized and terminated replications to make, and the runlength of each. It is well known that running a "long" simulation (i.e., taking the runlength large) will result in a sample mean that is "close" to the true mean performance; correspondingly, we will refer to a "long" simulation replication as a high-fidelity one and a "short" simulation replication as a low-fidelity one. The question of "whether a single long replication is preferable to several shorter ones" has been studied before; for example, see Kelton (1986), Whitt (1991), Alexopoulos and Goldsman (2004) and Grassmann (2016).

Relatively little attention has been given to metamodeling approaches for approximating a steady-state performance measure response surface across a design space of interest, with exceptions of Yang et al. (2008), Bekki et al. (2014) and Chen and Kim (2014), to name a few. In particular, an important yet underdeveloped topic is whether and how one can construct an adequate metamodel for approximating a mean response surface under a given simulation budget constraint, by utilizing steady-state simulation runs performed to controlled levels of fidelity at selected design points.

In this paper, we propose the stochastic co-kriging method (**SCK**) which extends co-kriging to the stochastic simulation setting for approximating a steady-state mean response surface. The remainder of this paper is organized as follows. Section 2 provides details on the framework for SCK, including stipulated assumptions, prediction and parameter estimation. Section 3 reviews a selection of correlation-based methods for steady-state variance estimation using outputs from within a single simulation replication. Section 4 presents a numerical example to demonstrate the competitive performance of SCK relative to stochastic kriging (Ankenman et al. 2010). Section 5 concludes this paper.

2 STOCHASTIC CO-KRIGING FOR SIMULATION METAMODELING

2.1 Building a Stochastic Co-Kriging Metamodel

It is known that estimating steady-state mean parameter through simulation can be computationally expensive. Typically, the longer the simulation runlength is, the more accurate and precise the sample mean as an estimator becomes. Simulations with shorter runlengths can be faster to run, yet the resulting estimators are less accurate and precise. In practice, we may consider running simulations using L different runlengths, which can be thought of as running simulations to L different levels of fidelity. For ease of exposition, we restrict our discussion to two levels of fidelity in this paper.

Let $\mathbf{D}_1$ and $\mathbf{D}_2$ represent, respectively, the design-point sets in $\mathscr{X} \subset \mathbb{R}^d$ for running the low- and high-fidelity simulation runs. More specifically, let $\mathbf{D}_1 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k_1}\}$ and $\mathbf{D}_2 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k_2}\}$, such that $\mathbf{D}_1 = \mathbf{D}_2 \cup \{\mathbf{x}_{k_2+1}, \mathbf{x}_{k_2+1}, \ldots, \mathbf{x}_{k_1}\}$. At design point $\mathbf{x}_i$ in $\mathbf{D}_1$ (for $i = 1,2,\ldots,k_1$), we perform $n_i^{\{1\}}$ low-fidelity simulation replications and generate independent and identically distributed (i.i.d.) simulation outputs $\{\mathscr{Y}_j^{\{1\}}(\mathbf{x}_i)\}_{j=1}^{n_i^{\{1\}}}$. Specifically, the jth low-fidelity simulation replication has a simulation runlength of $s_i^{\{1\}}$ (in terms of run time or number of more basic simulation outputs) which produces the low-fidelity simulation output $\mathscr{Y}_j^{\{1\}}(\mathbf{x}_i)$, for $j = 1,2,\ldots,n_i^{\{1\}}$. On the other hand, at design point $\mathbf{x}_i$ in $\mathbf{D}_2$ (for $i = 1,2,\ldots,k_2$), we perform $n_i^{\{2\}}$ high-fidelity simulation replications and generate i.i.d. simulation outputs $\{\mathscr{Y}_j^{\{2\}}(\mathbf{x}_i)\}_{j=1}^{n_i^{\{2\}}}$. That is, the jth high-fidelity simulation replication has a runlength of $s_i^{\{2\}}$ which produces the high-fidelity simulation output $\mathscr{Y}_j^{\{2\}}(\mathbf{x}_i)$. Furthermore, we assume $s_i^{\{2\}} \gg s_i^{\{1\}}$, for $i = 1,2,\ldots,k_2$. Hence, $\mathbf{D}_2$ denotes the set of design points where more simulation efforts are expended.

We next extend the co-kriging metamodeling methodology to the stochastic simulation setting for the purpose of studying steady-state simulation experiments. The mathematical structure of co-kriging

is expanded to encompass heterogeneous simulation output variances, which are referred to as intrinsic variability in Ankenman et al. (2010).

The low-accuracy performance measure estimator $\bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i)$ at design point $\mathbf{x}_i \in \mathbf{D}_1$ can be modeled as

$$\begin{aligned} \bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i) &= \frac{1}{n_i^{\{1\}}} \sum_{j=1}^{n_i^{\{1\}}} \mathscr{Y}_j^{\{1\}}(\mathbf{x}_i) \\ &= \mathsf{Y}^{\{1\}}(\mathbf{x}_i) + \bar{\zeta}^{\{1\}}(\mathbf{x}_i) \quad i = 1,2,\ldots,k_1, \end{aligned} \tag{1}$$

where $\mathsf{Y}^{\{1\}}(\mathbf{x}_i)$ denotes the unknown true mean of $\bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i)$, and $\bar{\zeta}^{\{1\}}(\mathbf{x}_i) = \sum_{j=1}^{n_i^{\{1\}}} \zeta_j^{\{1\}}(\mathbf{x}_i)/n_i^{\{1\}}$ denotes the simulation error in the estimator $\bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i)$. Notice that the $\zeta_j^{\{1\}}(\mathbf{x}_i)$'s represent the i.i.d. simulation errors with zero mean and variance $\mathsf{V}^{\{1\}}(\mathbf{x}_i)$. Hence, $\mathrm{Var}(\bar{\zeta}^{\{1\}}(\mathbf{x}_i)) = \mathsf{V}^{\{1\}}(\mathbf{x}_i)/n_i^{\{1\}}$ and it decreases with the number of replications applied at $\mathbf{x}_i$. We note that $\mathsf{V}^{\{1\}}(\mathbf{x}_i)$ measures the variance of the simulation output from each low-fidelity simulation replication, and it decreases with the simulation runlength $s_i^{\{1\}}$. If replications are available at $\mathbf{x}_i$ (i.e., $n_i^{\{1\}} > 1$), then $\mathsf{V}^{\{1\}}(\mathbf{x}_i)$ can be estimated by the sample variance $\widehat{\mathsf{V}}^{\{1\}}(\mathbf{x}_i)$ obtained at $\mathbf{x}_i \in \mathbf{D}_1$,

$$\widehat{\mathsf{V}}^{\{1\}}(\mathbf{x}_i) = \frac{1}{n_i^{\{1\}} - 1} \sum_{j=1}^{n_i^{\{1\}}} (\mathscr{Y}_j^{\{1\}}(\mathbf{x}_i) - \bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i))^2, \quad i = 1,2,\ldots,k_1.$$

We provide some further details on $\mathsf{Y}^{\{1\}}(\mathbf{x}_i)$ which can be described as follows:

$$\mathsf{Y}^{\{1\}}(\mathbf{x}_i) = \mathbf{f}_1(\mathbf{x}_i)^\top \beta_1 + \mathsf{M}_1(\mathbf{x}_i),$$

where β_1 is a $p_1 \times 1$ vector of parameters and $\mathbf{f}_1(\cdot)$ is a vector of known regression functions of compatible dimensions. As treated in the design and analysis of deterministic computer experiments literature (Santner et al. 2003), we assume that $\mathsf{M}_1(\cdot)$ is a mean-zero stationary Gaussian random field. There exists a spatial correlation function $\mathscr{R}_1(\cdot;\theta_1)$ that measures the correlation of the values of $\mathsf{M}_1(\mathbf{x}_i)$ and $\mathsf{M}_1(\mathbf{x}_\ell)$. This correlation is determined by the distance between $\mathbf{x}_i$ and $\mathbf{x}_\ell$ measured along each of the d dimensions, and the $d \times 1$ parameter vector $\theta_1 = (\theta_{11}, \theta_{12}, \ldots, \theta_{1d})^\top$ controls how quickly the spatial correlation diminishes as the two points become farther apart in each direction. Commonly used correlation functions include the Gaussian correlation function, Matérn correlation functions, and the exponential correlation function (see Chapter 4 of Rasmussen and Williams 2006); we choose to use the popular Gaussian correlation function $\mathscr{R}_1(\mathbf{x}_i, \mathbf{x}_\ell; \theta_1) = \exp\left(-\sum_{r=1}^d \theta_{1r}(x_{ir} - x_{\ell r})^2\right)$ in this paper. Given a correlation function, the implied covariance function is given by

$$\mathrm{Cov}(\mathsf{M}_1(\mathbf{x}_i), \mathsf{M}_1(\mathbf{x}_\ell)) = \tau_1^2 \mathscr{R}_1(\mathbf{x}_i, \mathbf{x}_\ell; \theta_1) \ , \tag{2}$$

where τ_1^2 denotes the variance of $\mathsf{M}_1(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}$.

On the other hand, we model the high-accuracy performance measure estimator $\bar{\mathscr{Y}}^{\{2\}}(\mathbf{x}_i)$ obtained at $\mathbf{x}_i \in \mathbf{D}_2$ as follows

$$\begin{aligned} \bar{\mathscr{Y}}^{\{2\}}(\mathbf{x}_i) &= \frac{1}{n_i^{\{2\}}} \sum_{j=1}^{n_i^{\{2\}}} \mathscr{Y}_j^{\{2\}}(\mathbf{x}_i) \\ &= \mathsf{Y}^{\{2\}}(\mathbf{x}_i) + \bar{\zeta}^{\{2\}}(\mathbf{x}_i), \\ &= \rho \mathsf{Y}^{\{1\}}(\mathbf{x}_i) + \delta(\mathbf{x}_i) + \bar{\zeta}^{\{2\}}(\mathbf{x}_i), \quad i = 1,2,\ldots,k_2, \end{aligned} \tag{3}$$

where $\mathsf{Y}^{\{2\}}(\mathbf{x}_i)$ represents the true mean of $\bar{\mathscr{Y}}^{\{2\}}(\mathbf{x}_i)$, $\delta(\mathbf{x}_i)$ denotes the difference between $\mathsf{Y}^{\{2\}}(\mathbf{x}_i)$ and $\rho\mathsf{Y}^{\{1\}}(\mathbf{x}_i)$ on which we will elaborate later. Notice that the $\zeta_j^{\{2\}}(\mathbf{x}_i)$'s denote the i.i.d. simulation errors with zero mean and variance $\mathsf{V}^{\{2\}}(\mathbf{x}_i)$, and $\bar{\zeta}^{\{2\}}(\mathbf{x}_i) := \sum_{j=1}^{n_i^{\{2\}}} \zeta_j^{\{2\}}(\mathbf{x}_i)/n_i^{\{2\}}$ denotes the average simulation error across the $n_i^{\{2\}}$ simulation replications at $\mathbf{x}_i$. Notice that $\mathrm{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_i)) = \mathsf{V}^{\{2\}}(\mathbf{x}_i)/n_i^{\{2\}}$ and it decreases with the number of replications $n_i^{\{2\}}$ applied at $\mathbf{x}_i$. Here $\mathsf{V}^{\{2\}}(\mathbf{x}_i)$ represents the variance of the simulation output generated from each high-fidelity simulation replication, and it decreases with the simulation runlength $s_i^{\{2\}}$. If replications are available (i.e., $n_i^{\{2\}} > 1$), then $\mathsf{V}^{\{2\}}(\mathbf{x}_i)$ can be estimated by the sample variance $\widehat{\mathsf{V}}^{\{2\}}(\mathbf{x}_i)$ obtained at $\mathbf{x}_i$ in a similar fashion as given in (2.1). We provide more details on estimation of $\mathsf{V}^{\{2\}}(\mathbf{x}_i)$ from a single high-fidelity simulation replication in Subsection 3.

We note that the model given in (3) relies on the following Markov property about true mean performance values implied by two-fidelity levels of simulation runs as introduced by Kennedy and O'Hagan (2000):

$$\mathrm{Cov}\left(\mathsf{Y}^{\{2\}}(\mathbf{x}), \mathsf{Y}^{\{1\}}(\widetilde{\mathbf{x}}) | \mathsf{Y}^{\{1\}}(\mathbf{x})\right) = 0, \tag{4}$$

for all $\mathbf{x} \neq \widetilde{\mathbf{x}}$. This property essentially states that if the true mean performance value implied by a *low-fidelity* simulation run at $\mathbf{x}$ is known, then we can learn no more about the true mean performance value implied from a *high-fidelity* simulation run at $\mathbf{x}$ from knowing any mean performance value of a *low-fidelity* simulation run at $\widetilde{\mathbf{x}}$ for $\widetilde{\mathbf{x}} \neq \mathbf{x}$.

We further model the difference term $\delta(\mathbf{x}_i)$ specified in (3) as

$$\delta(\mathbf{x}_i) = \mathbf{f}_2(\mathbf{x}_i)^\top \beta_2 + \mathsf{M}_2(\mathbf{x}_i), \tag{5}$$

where β_2 is a vector of unknown parameters, $\mathbf{f}_2(\cdot)$ is a vector of known regression functions of compatible dimensions and $\mathsf{M}_2(\cdot)$ is a stationary Gaussian process with mean zero, and covariance function $\mathrm{Cov}\left(\mathsf{M}_2(\mathbf{x}_i), \mathsf{M}_2(\mathbf{x}_\ell)\right) = \tau_\delta^2 \mathscr{R}_\delta(\mathbf{x}_i, \mathbf{x}_\ell; \theta_\delta)$. Notice that the discussion given for the spatial correlation function and hyperparameters for $\mathsf{M}_1(\cdot)$ applies to the spatial correlation function $\mathscr{R}_\delta(\cdot,\cdot;\theta_\delta)$ and the hyperparameters τ_δ and θ_δ for $\mathsf{M}_2(\mathbf{x}_i)$ here .

The true quantity of interest in our context, $\mathsf{Y}(\mathbf{x})$, can be a steady-state distribution parameter such as the steady-state mean of a random quantity of interest at $\mathbf{x}$. In spite that neither of the low- and high-fidelity point estimators, $\bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i)$ and $\bar{\mathscr{Y}}^{\{2\}}(\mathbf{x}_i)$, is unbiased for $\mathsf{Y}(\mathbf{x})$ (or equivalently, $\mathsf{Y}^{\{1\}}(\mathbf{x}_i) \neq \mathsf{Y}(\mathbf{x}_i)$ and $\mathsf{Y}^{\{2\}}(\mathbf{x}_i) \neq \mathsf{Y}(\mathbf{x}_i)$), the simulation runlength ($s_i^{\{1\}}$ or $s_i^{\{2\}}$) applied at a design point determines the bias and variance of the point estimator obtained. In particular, $|\mathrm{Bias}[\bar{\mathscr{Y}}^{\{2\}}(\mathbf{x}_i)]| \leq |\mathrm{Bias}[\bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i)]|$ and $\mathsf{V}^{\{2\}}(\mathbf{x}_i) \leq \mathsf{V}^{\{1\}}(\mathbf{x}_i)$, for $\mathbf{x}_i \in \mathbf{D}_2$.

2.2 Prediction by Stochastic Co-Kriging

Assuming that all hyperparameters are given, we now perform the stochastic co-kriging prediction of the expected high-fidelity response at a prediction point $\mathbf{x}_0$. Notice that standard results indicate that $\left(\mathsf{Y}^{\{2\}}(\mathbf{x}_0), \bar{\mathscr{Y}}^\top\right)^\top$ follow a multivariate normal distribution (Kennedy and O'Hagan 2000), where $\bar{\mathscr{Y}} = \left(\bar{\mathscr{Y}}^{\{1\}}, \bar{\mathscr{Y}}^{\{2\}}\right)^\top$ and $\bar{\mathscr{Y}}^{\{i\}} = \left(\bar{\mathscr{Y}}^{\{i\}}(\mathbf{x}_1), \bar{\mathscr{Y}}^{\{i\}}(\mathbf{x}_2), \ldots, \bar{\mathscr{Y}}^{\{i\}}(\mathbf{x}_{k_i})\right)^\top$ for $i = 1, 2$. In particular, the conditional distribution of $\mathsf{Y}^{\{2\}}(\mathbf{x}_0)$ given $\bar{\mathscr{Y}}$ is also normal with the mean function given by

$$\widehat{\mathsf{Y}}^{\{2\}}(\mathbf{x}_0) = \mathbf{f}(\mathbf{x}_0)^\top \widehat{\beta} + \mathbf{c}(\mathbf{x}_0)^\top \Sigma^{-1} (\bar{\mathscr{Y}} - \mathbf{F}\widehat{\beta}) \tag{6}$$

where $\mathbf{f}(\mathbf{x}_0)^\top = (\rho \mathbf{f}_1(\mathbf{x}_0)^\top, \mathbf{f}_2(\mathbf{x}_0)^\top)$, and

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1)^\top & 0 \\ \vdots & \vdots \\ \mathbf{f}_1(\mathbf{x}_{k_1})^\top & 0 \\ \rho \mathbf{f}_1(\mathbf{x}_1)^\top & \mathbf{f}_2(\mathbf{x}_1)^\top \\ \vdots & \vdots \\ \rho \mathbf{f}_1(\mathbf{x}_{k_2})^\top & \mathbf{f}_2(\mathbf{x}_{k_2})^\top \end{bmatrix}, \tag{7}$$

$$\widehat{\beta} = \left(\widehat{\beta}_1^\top, \widehat{\beta}_2^\top\right)^\top = \left(\mathbf{F}^\top \Sigma^{-1} \mathbf{F}\right)^{-1} \mathbf{F}^\top \Sigma^{-1} \bar{\mathscr{Y}}, \tag{8}$$

and $\Sigma = \Sigma_{\mathsf{M}} + \Sigma_\varepsilon$, and $\mathbf{c}(\mathbf{x}_0)$ denotes the following $(k_1 + k_2) \times 1$ covariance vector

$$\mathbf{c}(\mathbf{x}_0) = \left(\rho \tau_1^2 \mathscr{R}_1(\mathbf{D}_1, \mathbf{x}_0; \theta_1)^\top \qquad \rho^2 \tau_1^2 \mathscr{R}_1(\mathbf{D}_2, \mathbf{x}_0; \theta_1)^\top + \tau_\delta^2 \mathscr{R}_\delta(\mathbf{D}_2, \mathbf{x}_0; \theta_\delta)^\top\right)^\top,$$

where $\mathscr{R}_1(\mathbf{D}_i, \mathbf{x}_0; \theta_1)$ denotes the $k_i \times 1$ vector of spatial correlations between $\mathsf{Y}^{\{1\}}(\mathbf{x}_0)$ and $\mathsf{Y}^{\{1\}}(\mathbf{x}_\ell)$, for $\ell = 1, 2, \ldots, k_i$, $i = 1, 2$; $\mathscr{R}_\delta(\mathbf{D}_2, \mathbf{x}_0; \theta_\delta)$ denotes the $k_2 \times 1$ vector of spatial correlations between $\mathsf{Y}^{\{2\}}(\mathbf{x}_0)$ and $\mathsf{Y}^{\{2\}}(\mathbf{x}_\ell)$ for $\ell = 1, 2, \ldots, k_2$. Furthermore, $\Sigma = \Sigma_{\mathsf{M}} + \Sigma_\varepsilon$, and

$$\Sigma_{\mathsf{M}} = \begin{pmatrix} \Sigma_{\mathsf{M}}^{11} & \Sigma_{\mathsf{M}}^{12} \\ \left(\Sigma_{\mathsf{M}}^{12}\right)^\top & \Sigma_{\mathsf{M}}^{22} \end{pmatrix} = \begin{pmatrix} \tau_1^2 \mathscr{R}_1(\mathbf{D}_1, \mathbf{D}_1; \theta_1) & \rho \tau_1^2 \mathscr{R}_1(\mathbf{D}_1, \mathbf{D}_2; \theta_1) \\ \rho \tau_1^2 \mathscr{R}_1(\mathbf{D}_1, \mathbf{D}_2; \theta_1)^\top & \rho^2 \tau_1^2 \mathscr{R}_1(\mathbf{D}_2, \mathbf{D}_2; \theta_1) + \tau_\delta^2 \mathscr{R}_\delta(\mathbf{D}_2, \mathbf{D}_2; \theta_\delta). \end{pmatrix}$$

Notice that the notation $\mathscr{R}_1(\mathbf{D}_1, \mathbf{D}_2; \theta_1)$ denotes the matrix of correlations between the values of $\mathsf{Y}^{\{1\}}(\cdot)$ at design points in $\mathbf{D}_1$ and $\mathbf{D}_2$, with its (i, j)th entry given by $\mathscr{R}_1(\mathbf{x}_i, \mathbf{x}_j; \theta_1)$ for all $\mathbf{x}_i \in \mathbf{D}_1$ and $\mathbf{x}_j \in \mathbf{D}_2$. The other notation such as $\mathscr{R}_1(\mathbf{D}_1, \mathbf{D}_1; \theta_1)$ and $\mathscr{R}_\delta(\mathbf{D}_2, \mathbf{D}_2; \theta_\delta)$ is defined in a similar fashion.

The intrinsic variance-covariance matrix of $\bar{\mathscr{Y}}$ is

$$\Sigma_\varepsilon = \begin{pmatrix} \Sigma_\varepsilon^{11} & \Sigma_\varepsilon^{12} \\ \left(\Sigma_\varepsilon^{12}\right)^\top & \Sigma_\varepsilon^{22} \end{pmatrix},$$

where Σ_ε^{ii} is the $k_i \times k_i$ variance-covariance matrix of $\bar{\mathscr{Y}}^{\{i\}}$ for $i = 1, 2$; and Σ_ε^{12} is the $k_1 \times k_2$ covariance matrix of $\bar{\mathscr{Y}}^{\{1\}}$ and $\bar{\mathscr{Y}}^{\{2\}}$. Specifically,

$$\Sigma_\varepsilon^{11} = \operatorname{diag}\left(\operatorname{Var}(\bar{\zeta}^{\{1\}}(\mathbf{x}_1)), \ldots, \operatorname{Var}(\bar{\zeta}^{\{1\}}(\mathbf{x}_{k_1}))\right) = \operatorname{diag}\left(\mathsf{V}^{\{1\}}(\mathbf{x}_1)/n_1^{\{1\}}, \ldots, \operatorname{Var}(\mathsf{V}^{\{1\}}(\mathbf{x}_{k_1})/n_{k_1}^{\{1\}})\right),$$

$$\Sigma_\varepsilon^{22} = \operatorname{diag}\left(\operatorname{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_1)), \ldots, \operatorname{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_{k_2}))\right) = \operatorname{diag}\left(\mathsf{V}^{\{2\}}(\mathbf{x}_1)/n_1^{\{2\}}, \ldots, \operatorname{Var}(\mathsf{V}^{\{2\}}(\mathbf{x}_{k_2})/n_{k_2}^{\{2\}})\right),$$

$$\Sigma_\varepsilon^{12} = \begin{pmatrix} \operatorname{diag}(\operatorname{Cov}(\bar{\zeta}^{\{1\}}(\mathbf{x}_1), \bar{\zeta}^{\{2\}}(\mathbf{x}_1)), \ldots, \operatorname{Cov}(\bar{\zeta}^{\{1\}}(\mathbf{x}_{k_2}), \bar{\zeta}^{\{2\}}(\mathbf{x}_{k_2}))) \\ \mathbf{0}_{(k_1 - k_2) \times k_2} \end{pmatrix},$$

where $\mathbf{0}_{(k_1 - k_2) \times k_2}$ represents a $(k_1 - k_2) \times k_2$ matrix of zeros, and for $i = 1, 2, \ldots, k_2$,

$$\operatorname{Cov}(\bar{\zeta}^{\{1\}}(\mathbf{x}_i), \bar{\zeta}^{\{2\}}(\mathbf{x}_i)) = \operatorname{Cov}(\zeta_j^{\{1\}}(\mathbf{x}_i), \zeta_j^{\{2\}}(\mathbf{x}_i)) / \max\{n_i^{\{1\}}, n_i^{\{2\}}\}.$$

The conditional prediction variance follows as

$$\operatorname{Var}(\widehat{\mathsf{Y}}^{\{2\}}(\mathbf{x}_0)) = \tau_\delta^2 + \rho^2 \tau_1^2 - \mathbf{c}(\mathbf{x}_0)^\top \Sigma^{-1} \mathbf{c}(\mathbf{x}_0) + \eta(\mathbf{x}_0)^\top \left(\mathbf{F}^\top \Sigma^{-1} \mathbf{F}\right)^{-1} \eta(\mathbf{x}_0), \tag{9}$$

where $\eta(\mathbf{x}_0) = \mathbf{f}(\mathbf{x}_0) - \mathbf{c}(\mathbf{x}_0)^\top \Sigma^{-1}\mathbf{F}$.

We note that the predictor given in (6) can be used as a cheap approximation to the mean function value implied by high-fidelity simulation runs at a given prediction point $\mathbf{x}_0 \in \mathscr{X}$. Provided that high-fidelity simulations have been performed at enough design points, (6) should be more accurate than that given based on low-fidelity simulation runs. The conditional prediction variance (9) can be used to measure the prediction uncertainty associated with (6).

2.3 Estimating the Model Hyperparameters

Given that the parameter vector β is estimated by $\widehat{\beta}$ given in (8), below we consider the estimation of model hyperparameters. As a result of the choice of design-point locations in $\mathbf{D}_1$ and those in $\mathbf{D}_2 \subset \mathbf{D}_1$ and the Markov property, we can estimate the parameters $(\tau_1^2, \theta_1^\top)$ separately from $(\rho, \tau_\delta^2, \theta_\delta^\top)$ following a similar argument as given by Kennedy and O'Hagan (2000).

Conditional on $(\tau_1^2, \theta_1^\top)$, the distribution of $\bar{\mathscr{Y}}^{\{1\}}$ is normal and the log-likelihood of $\bar{\mathscr{Y}}^{\{1\}}$ can be written as

$$\ln \mathscr{L}(\tau_1^2, \theta_1^\top) = -\left(\frac{k_1}{2}\ln(2\pi) + \frac{1}{2}\ln\left(\det\left(\Sigma^{11}\right)\right) + \frac{1}{2}\left(\bar{\mathscr{Y}}^{\{1\}} - \mathbf{F}_1\widehat{\beta}_1\right)^\top \left(\Sigma^{11}\right)^{-1}\left(\bar{\mathscr{Y}}^{\{1\}} - \mathbf{F}_1\widehat{\beta}_1\right)\right), \quad (10)$$

where

$$\mathbf{F}_1 = \begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1)^\top \\ \vdots \\ \mathbf{f}_1(\mathbf{x}_{k_1})^\top \end{bmatrix},$$

$\Sigma^{11} = \tau_1^2 \mathscr{R}_1(\mathbf{D}_1, \mathbf{D}_1; \theta_1) + \Sigma_\varepsilon^{11}$. First, we need to estimate Σ_ε^{11} and replace it by its estimator in (10). Then estimates of τ_1^2 and θ_1 can be obtained by suitable optimization routines such as those available in Matlab.

We write the vector of differences between the two point estimators built on low- and high-fidelity simulation runs as

$$\widetilde{\delta} = \bar{\mathscr{Y}}^{\{2\}} - \rho\bar{\mathscr{Y}}^{\{1\}} = \left(\widetilde{\delta}(\mathbf{x}_1), \widetilde{\delta}(\mathbf{x}_2), \ldots, \widetilde{\delta}(\mathbf{x}_{k_2})\right)^\top.$$

It follows from (1) and (3) and the description given in Subsection 2.1 that

$$\widetilde{\delta}(\mathbf{x}_i) = \bar{\mathscr{Y}}^{\{2\}}(\mathbf{x}_i) - \rho\bar{\mathscr{Y}}^{\{1\}}(\mathbf{x}_i) = \delta(\mathbf{x}_i) + \bar{\zeta}^{\{2\}}(\mathbf{x}_i) - \rho\bar{\zeta}^{\{1\}}(\mathbf{x}_i), \quad \text{for } \mathbf{x}_i \in \mathbf{D}_2.$$

Conditional on $(\rho, \tau_\delta^2, \theta_\delta^\top)$, the distribution of $\widetilde{\delta}$ is normal and the log-likelihood of $\widetilde{\delta}$ can be written as

$$\ln \mathscr{L}(\rho, \tau_\delta^2, \theta_\delta^\top) = -\left(\frac{k_2}{2}\ln(2\pi) + \frac{1}{2}\ln\left(\det\left(\Sigma^{22}\right)\right) + \frac{1}{2}\left(\widetilde{\delta} - \mathbf{F}_\delta\widehat{\beta}_2\right)^\top \left(\Sigma^{22}\right)^{-1}\left(\widetilde{\delta} - \mathbf{F}_\delta\widehat{\beta}_2\right)\right), \quad (11)$$

where

$$\mathbf{F}_\delta = \begin{bmatrix} \mathbf{f}_2(\mathbf{x}_1)^\top \\ \vdots \\ \mathbf{f}_2(\mathbf{x}_{k_2})^\top \end{bmatrix},$$

$\Sigma^{22} = \tau_\delta^2 \mathscr{R}_\delta(\mathbf{D}_2, \mathbf{D}_2; \theta_\delta) + \Sigma_\varepsilon^{\widetilde{\delta}}$, with $\Sigma_\varepsilon^{\widetilde{\delta}} = \mathrm{diag}\left(\mathrm{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_1) - \rho\bar{\zeta}^{\{1\}}(\mathbf{x}_1)), \ldots, \mathrm{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_{k_2}) - \rho\bar{\zeta}^{\{1\}}(\mathbf{x}_{k_2}))\right)$, and for $\mathbf{x}_i \in D_2$,

$$\begin{aligned} \mathrm{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_i) - \rho\bar{\zeta}^{\{1\}}(\mathbf{x}_i)) &= \mathrm{Var}(\bar{\zeta}^{\{2\}}(\mathbf{x}_i)) + \rho^2\mathrm{Var}(\bar{\zeta}^{\{1\}}(\mathbf{x}_i)) - 2\rho\mathrm{Cov}(\bar{\zeta}^{\{2\}}(\mathbf{x}_i), \bar{\zeta}^{\{1\}}(\mathbf{x}_i)) \\ &= \mathrm{V}^{\{2\}}(\mathbf{x}_i)/n_i^{\{2\}} + \rho^2\mathrm{V}^{\{1\}}(\mathbf{x}_i)/n_i^{\{1\}} - 2\rho\mathrm{Cov}(\zeta_j^{\{1\}}(\mathbf{x}_i), \zeta_j^{\{2\}}(\mathbf{x}_i))/\max\{n_i^{\{1\}}, n_i^{\{2\}}\}. \end{aligned}$$

First, we need to estimate $\Sigma_\varepsilon^{\widetilde{\delta}}$ in Σ^{22} and replace it by its estimator in (11). Estimates of ρ, τ_δ^2 and θ_2 can be obtained by suitable optimization routines subsequently.

3 METHODS FOR STEADY-STATE VARIANCE ESTIMATION

In this section we review a small selection of methods for steady-state variance estimation. These methods facilitate the application of SCK using outputs from within a single high-fidelity simulation replication. We will concentrate on discrete-time processes (continuous-time processes can be handled in a similar manner). Recall that at each low-fidelity design point $\mathbf{x}_i$ in $\mathbf{D}_1$, we run $n_i^{\{1\}}$ independent simulation replications and generate i.i.d. outputs $\{\mathscr{Y}_j^{\{1\}}(\mathbf{x}_i)\}_{j=1}^{n_i^{\{1\}}}$. The output $\mathscr{Y}_j^{\{1\}}(\mathbf{x}_i)$ generated on the jth simulation replication is considered as the sample mean of $s_i^{\{1\}}$ basic outputs, i.e., $\mathscr{Y}_1^{\{1\}}(\mathbf{x}_i) = \sum_{t=1}^{s_i^{\{1\}}} Y_t(\mathbf{x}_i)/s_i^{\{1\}}$. On the other hand, at each high-fidelity design point $\mathbf{x}_i \in \mathbf{D}_2$, a single simulation replication is performed with a runlength much longer than that of a low-fidelity simulation replication, i.e., $s_i^{\{2\}} >> s_i^{\{1\}}$, and produces a single point estimate $\mathscr{Y}_1^{\{2\}}(\mathbf{x}_i) = \sum_{t=1}^{s_i^{\{2\}}} Y_t(\mathbf{x}_i)/s_i^{\{2\}}$, the sample mean of $s_i^{\{2\}}$ basic outputs at $\mathbf{x}_i$.

Given a single long simulation replication at each high-fidelity design point $\mathbf{x}_i \in \mathbf{D}_2$, we next consider estimating the variance of the sample mean, $\mathsf{V}^{\{2\}}(\mathbf{x}_i)$, via some selected correlation-based methods; see details from, for example, Alexopoulos and Goldsman (2004), Goldsman and Nelson (2006), Alexopoulos et al. (2007) and Alexopoulos et al. (2007). For ease of exposition, we omit the design point $\mathbf{x}_i$ from our notation and further denote $\mathsf{V}^{\{2\}}(\mathbf{x}_i)$ by V.

Nonoverlapping Batch Mean Variance Estimator (NBM)

Suppose that each high fidelity run has a runlength of $s^{\{2\}} = mb$, and the simulation outputs, $Y_1, Y_2, \ldots, Y_{s^{\{2\}}}$, can be divided into b contiguous, nonoverlapping batches of outputs, each of batch size m. That is, the ith batch is consisted of observations $Y_{(i-1)m+1}, Y_{(i-1)m+2}, \ldots, Y_{im}$, for $i = 1, 2, \ldots, b$. The NBM estimator for V is given by

$$\widehat{\mathsf{V}}_{\text{NBM}}^{\{2\}} = \frac{m}{(b-1)s^{\{2\}}} \sum_{i=1}^{b} (\bar{Y}_{i,m} - \bar{Y}_{s^{\{2\}}})^2,$$

where $\bar{Y}_{i,m} = m^{-1} \sum_{\ell=1}^{m} Y_{(i-1)m+\ell}$ for $i = 1, 2, \ldots, b$; and $\bar{Y}_{s^{\{2\}}} = \sum_{i=1}^{s^{\{2\}}} Y_i / s^{\{2\}}$.

The next two variance estimators are constructed from the following standardized time series (STS) based on the ith nonoverlapping batch of size m,

$$T_{i,m}(t) = \frac{\lfloor mt \rfloor \left(\bar{Y}_{i,m} - \bar{Y}_{i,\lfloor mt \rfloor} \right)}{\sqrt{\mathsf{V} m}}, \quad \text{for } t \in [0,1],$$

where $\lfloor \cdot \rfloor$ denotes the floor function and $\bar{Y}_{i,j} = j^{-1} \sum_{\ell=1}^{j} Y_{(i-1)m+\ell}$ denotes the jth cumulative sample mean for $j = 1, 2, \ldots, m$ from the ith batch, $i = 1, 2, \ldots, b$.

Nonoverlapping Batched Area Variance Estimator (NA)

We denote $A_i(f; m)$ as the weighted area estimator computed under the STS from the ith nonoverlapping batch,

$$A_i(f; m) = \left[m^{-1} \sum_{\ell=1}^{m} f\left(\frac{\ell}{m}\right) \mathsf{V}^{\frac{1}{2}} T_{i,m}\left(\frac{\ell}{m}\right) \right]^2, \quad \text{for } i = 1, 2, \ldots, b.$$

where $f(\cdot)$ is a weighting function and we adopt $f(t) = \sqrt{840}(3t^2 - 3t + 0.5)$ for $t \in [0,1]$ in Section 4 for numerical evaluations; see other weighting functions from, for instance, Goldsman and Nelson (2006). The NA estimator follows as

$$\widehat{\mathsf{V}}_{\text{NA}}^{\{2\}} = \frac{1}{bs^{\{2\}}} \sum_{i=1}^{b} A_i(f; m).$$

Nonoverlapping Batched Weighted Cramér-von Mises Estimator (NCvM)

We denote $C_i(g;m)$ as the weighted area under the STS from the ith nonoverlapping batch,

$$C_i(g;m) = m^{-1}\sum_{\ell=1}^{m} g\left(\frac{\ell}{m}\right) \mathsf{V} T_{i,m}^2\left(\frac{\ell}{m}\right), \quad \text{for } i = 1,2,\ldots,b.$$

where $g(\cdot)$ is a weighting function and we adopt $g(t) = -24 + 150t - 150t^2$ for $t \in [0,1]$ in Section 4 for numerical evaluations; other weight functions can be found from, for example, Goldsman and Nelson (2006). The NCvM estimator is given by

$$\widehat{\mathsf{V}}_{\text{NCvM}}^{\{2\}} = \frac{1}{bs^{\{2\}}}\sum_{i=1}^{b} C_i(g;m).$$

Overlapping Batch Mean Variance Estimator (OBM)

Suppose that we divide $Y_1, Y_2, \ldots, Y_{n_i^{\{2\}}}$ into $s^{\{2\}} - m + 1$ overlapping batches, each of size m. That is, the observations $Y_1, Y_2, \ldots, Y_m$ comprise the 1st batch, and $Y_2, Y_3, \ldots, Y_{m+1}$ form the 2nd batch, and so on. In general, $Y_i, Y_{i+1}, \ldots, Y_{i+m-1}$ form the ith batch, for $i = 1,2,\ldots,n_i^{\{2\}} - m + 1$. The OBM estimator can be given by

$$\widehat{\mathsf{V}}_{\text{OBM}}^{\{2\}} = \frac{s^{\{2\}} m}{(s^{\{2\}} - m + 1)(s^{\{2\}} - m)s^{\{2\}}} \sum_{i=1}^{s^{\{2\}}-m+1} \left(\bar{Y}_{i,m}^{O} - \bar{Y}_{s^{\{2\}}}\right)^2,$$

where $\bar{Y}_{i,m}^{O} = m^{-1}\sum_{\ell=0}^{m-1} Y_{i+\ell}$ for $i = 1,2,\ldots,s^{\{2\}} - m + 1$, and $\bar{Y}_{s^{\{2\}}} = \sum_{i=1}^{s^{\{2\}}} Y_i / s^{\{2\}}$.

The next two variance estimators are constructed from the following STS based on the ith overlapping batch of size m,

$$T_{i,m}^{O}(t) = \frac{\lfloor mt \rfloor \left(\bar{Y}_{i,m}^{O} - \bar{Y}_{i,\lfloor mt \rfloor}^{O}\right)}{\sqrt{\mathsf{V} m}}, \quad \text{for } t \in [0,1],$$

for $i = 1,2,\ldots,s^{\{2\}} - m + 1$, where $\bar{Y}_{i,j}^{O} = j^{-1}\sum_{\ell=0}^{j-1} Y_{i+\ell}$ for $i = 1,2,\ldots,s^{\{2\}} - m + 1$ and $j = 1,2,\ldots,m$.

Overlapping Batched Area Variance Estimator (OA)

We denote $A_i^{O}(f;m)$ as the weighted area estimator computed under the STS from the ith overlapping batch,

$$A_i^{O}(f;m) = \left[m^{-1}\sum_{\ell=1}^{m} f\left(\frac{\ell}{m}\right) \mathsf{V}^{\frac{1}{2}} T_{i,m}^{O}\left(\frac{\ell}{m}\right)\right]^2, \quad \text{for } i = 1,2,\ldots,s^{\{2\}} - m + 1,$$

where the weighting function f is the same as described for the NA estimator. The OA estimator then follows as

$$\widehat{\mathsf{V}}_{\text{OA}}^{\{2\}} = \frac{1}{(s^{\{2\}} - m + 1)s^{\{2\}}} \sum_{i=1}^{s^{\{2\}}-m+1} A_i^{O}(f;m).$$

Nonoverlapping Batched Weighted Cramér-von Mises Estimator (OCvM)

We denote $C_i^{O}(g;m)$ as the weighted area under the STS from the ith overlapping batch,

$$C_i^{O}(g;m) = m^{-1}\sum_{\ell=1}^{m} g\left(\frac{\ell}{m}\right)\left[\mathsf{V}^{\frac{1}{2}} T_{i,m}^{O}\left(\frac{\ell}{m}\right)\right]^2, \quad \text{for } i = 1,2,\ldots,s^{\{2\}} - m + 1.$$

where the weighting function g is the same as described for the NCvM estimator. The OCvM estimator then follows as

$$\widehat{\mathsf{V}}^{\{2\}}_{\mathrm{OCvM}} = \frac{1}{(s^{\{2\}}-m+1)s^{\{2\}}} \sum_{i=1}^{s^{\{2\}}-m+1} C_i^O(g;m).$$

4 AN M/M/1 QUEUE EXAMPLE

Consider simulating an M/M/1 queue with arrival rate 1 per time unit and service rate x per time unit with $x \in \mathscr{X} = [1.1, 2]$. It is well known from queueing theory that the mean steady-state waiting time in the queue is $\mathsf{Y}(x) = 1/\big(x(x-1)\big)$ (Whitt 1989), which is the function we intend to estimate. For each simulation experiment, a set of k equispaced design points are chosen from $\mathscr{X}$, with $x_1 = 1.1$ and $x_k = 2$. Each simulation replication (run) is *initialized either in empty state or steady state*, and the runlength T is specified by the number of simulated customers. We note that the value of T here determines the steady-state simulation fidelity level. The simulation output on a given replication is the sample-path average waiting time of the T customers simulated.

4.1 Experiment Setup

Two-fidelity-level simulation experiment. We consider two sets of two-fidelity-level simulation experiment, which share the common low-fidelity simulation runs and only differ in the high-fidelity simulation runs. The low-fidelity design-point set $\mathbf{D}_1$ consists of a grid of 25 equidistant design points in $[1.1, 2]$ with $x_1 = 1.1$ and $x_{25} = 2$. At each design point in $\mathbf{D}_1$, $n^{\{1\}} = 10$ simulation replications are applied and the runlength of each replication is 5000. The high-fidelity design-point set $\mathbf{D}_2 \subset \mathbf{D}_1$ consists of 4 design points, i.e., $\mathbf{D}_2 = \{x_1, x_9, x_{17}, x_{25}\}$. We consider the following two types of high-fidelity simulation runs:

1. **High-fidelity simulation with multiple replications:** At each design point in $\mathbf{D}_2$, $n^{\{2\}} = 4$ replications are applied with each replication having a runlength of $275,000$. Therefore, the simulation budget expended at each high fidelity design point is 1.1×10^6.
2. **High-fidelity simulation with a single replication :** At each design point in $\mathbf{D}_2$, a single simulation replication is applied with a runlength of 1.1×10^6.

Despite the difference in the two sets of high-fidelity simulation runs, we note that the resulting total simulation budget for the above two sets of two-fidelity-level simulation experiment stays the same which is 5.65×10^6.

Single-fidelity-level simulation experiment. We consider conducting a single-fidelity-level simulation experiment at the same set of design points as those in $\mathbf{D}_1$. Specifically, at each design point in $\mathbf{D}_1$, $n = 10$ replications are applied with each replication having a runlength of $22,600$. The total simulation budget is the same as that of the two sets of two-fidelity-level simulation experiment.

We consider the following metamodeling methods and compare their predictive performance: (1) stochastic kriging applied with the single-fidelity-level simulation experiment (**SK-1L**), (2) stochastic co-kriging applied with the two-fidelity simulation experiment in which high-fidelity simulations are replicated (**SCK-mH**), and (3) stochastic co-kriging applied with the two-fidelity simulation experiment in which a single high-fidelity simulation replication is used (**SCK-sH**). For implementing SCK-sH, we use the methods reviewed in Subsection 3 for estimating the variance of a point estimate using the individual waiting times generated from within a single long replication. Notice that for the sake of brevity, we omit the results obtained by SCK with OCvM and OA applied. An important decision in this context is to determine the batch size m to use for variance estimation. For discussions of appropriate batch sizes to use, see Nelson (2011), Song and Schmeiser (1995) and Song (1996), to name a few. In our implementation, we set the ratio of the runlength to the batch size $b = s^{\{2\}}/m$ to 20, 50, and 110 which corresponds to $m = 55,000$, $22,000$, $10,000$, respectively.

A grid of $K = 193$ equispaced check-points are chosen from $\mathscr{X}$ to evaluate predictive performance of stochastic co-kriging (SCK) and stochastic kriging (SK). The aforementioned two-fidelity-level and single-fidelity-level experiments are respectively executed for 100 independent macro-replications, and the predictive performance measure, the empirical root mean squared errors (ERMSE), is calculated as follows,

$$\text{ERMSE}_\ell = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(\widehat{\text{Y}}_\ell(x_i) - \text{Y}(x_i)\right)^2}, \quad \ell = 1,2,\ldots,100, \tag{12}$$

where $\widehat{\text{Y}}_\ell(\cdot)$ represents the prediction given by SCK or SK on the ℓth macro-replication.

4.2 Summary of Results

The ERMSEs obtained by SCK and SK from 100 macro-replications are summarized in Table 1. The value in each cell of Table 1 is the average ERMSE obtained across the 100 macro-replications; and the value in parentheses is the corresponding standard error. We observe that regardless of the initializing condition, SCK-mH outperforms SCK-sH and SK-1L and SK-1L performs the worst. The performances achieved by SCK-sH with different batch means methods applied is close to one another, and the ERMSEs obtained are relatively stable as the batch size increases from 10,000 to 55,000. In terms of experimental design for the two-level-fidelity simulation experiment, we observe that while keeping the low-fidelity simulation runs fixed, SCK seems to work better with a few moderately long simulation replications as compared to a single long simulation replication; and a lack of replications at high-fidelity design points may lead to loss of predictive accuracy achieved by SCK. Lastly, initializing a simulation run at steady state does not seem to make a significant impact on the performance achieved by SCK as opposed to initializing in empty state.

Table 1: Results for the M/M/1 queueing example.

Initialization	SK-1L	SCK-mH	SCK-sH + batch mean methods				
			batch size	NBM	NA	NCvM	OBM
Empty state	0.484 (0.008)	0.39 (0.01)	10,000	0.40 (0.02)	0.41 (0.02)	0.41 (0.02)	0.40 (0.02)
			22,000	0.40 (0.02)	0.41 (0.02)	0.41 (0.02)	0.40 (0.02)
			55,000	0.40 (0.02)	0.41 (0.02)	0.41 (0.02)	0.40 (0.02)
			batch size	NBM	NA	NCvM	OBM
Steady state	0.494 (0.007)	0.38 (0.01)	10,000	0.40 (0.01)	0.41 (0.01)	0.41 (0.01)	0.40 (0.01)
			22,000	0.40 (0.01)	0.41 (0.01)	0.41 (0.01)	0.40 (0.01)
			55,000	0.40 (0.01)	0.41 (0.01)	0.41 (0.01)	0.40 (0.01)

5 CONCLUSIONS

In summary, we have presented the stochastic co-kriging methodology (SCK) for approximating an steady-state mean response surface based on outputs from both long and short simulation replications performed at selected design points. We have provided details on how to construct an SCK metamodel, perform parameter estimation, and make prediction via SCK. From a deign of simulation experiments perspective, metamodels reduce the computational cost of exploring large regions of the design space by replacing replicated long simulations required to obtain accurate steady-state mean parameter estimates. However, it is well known that a substantial computational effort is involved in performing long steady-state simulations to build metamodels. Using SCK proposed in this paper, with the same computational effort expended, it is possible

to improve the accuracy of the metamodels obtained from the relatively short simulation replications, by supplementing the outputs from these simulations with outputs from long simulation replications performed at only a few design points. Therefore, it is possible to explore a design space with enhanced metamodels that are more accurate than metamodels based entirely on short simulation replications but less computationally expensive than metamodels based exclusively on long simulation replications.

We have shown the promise of using SCK for approximating a mean response surface using simulation runs performed to two levels of fidelity. This method can be extendable to multiple levels and there exist many other types of wisdom that may be incorporated into simulation experimental designs for SCK, such as approximation results from queueing theory (e.g., Whitt 1989 and Whitt 2006). Future research topics include investigating design-point sets for performing simulations with different levels of fidelity and seeking suitable simulation budget allocation rules when a fixed computational budget is given.

ACKNOWLEDGMENTS

The work of Xi Chen was partially supported by the Virginia Tech ICTAS Junior Faculty Award.

REFERENCES

Alexopoulos, C., N. T. Argon, D. Goldsman, N. M. Steiger, G. Tokol, and J. R. Wilson. 2007. "Efficient Computation of Overlapping Variance Estimators for Simulation". *INFORMS Journal on Computing* 29:314–327.

Alexopoulos, C., N. T. Argon, D. Goldsman, G. Tokol, and J. R. Wilson. 2007. "Overlapping Variance Estimators for Simulation". *Operations Research* 55 (6): 1090–1103.

Alexopoulos, C., and D. Goldsman. 2004. "To Batch or Not to Batch?". *ACM Transactions on Modeling and Computer Simulation* 14:76–114.

Ankenman, B. E., B. L. Nelson, and J. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58:371–382.

Argon, N. T., S. Andradóttir, C. Alexopoulos, and D. Goldsman. 2013. "Steady-State Simulation With Replication-Dependent Initial Transients: Analysis and Examples". *INFORMS Journal on Computing* 25:177–191.

Bekki, J. M., X. Chen, and D. Batur. 2014. "Steady-State Quantile Parameter Estimation: An Empirical Comparison of Stochastic Kriging and Quantile Regression". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 3880–3891. 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.

Cressie, N. A. C. 2015. *Statistics for Spatial Data*. Revised ed. New Jersey: John Wiley & Sons.

Forrester, A. I. J., A. Sóbester, and A. J. Keane. 2007. "Multi-Fidelity Optimization via Surrogate Modelling". In *Proceedings of the Royal Society A*, edited by J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 3251–3269.

Goldsman, D., and B. L. Nelson. 2006. "Correlation-Based Methods for Output Analysis". In *Simulation*, edited by S. G. Henderson and B. L. Nelson, Handbooks in Operations Research and Management Science Vol. 13, 455–475. Elsevier.

Grassmann, W. 2016. "Multiple Runs in the Simulation of Stochastic Systems Can Improve the Estimates of Equilibrium Expectations". In *Proceedings of Simulation and Modeling Methodologies, Technologies and Applications: International Conference (SIMULTECH 2015)*, edited by M. S. Obaidat, J. Kacprzyk, T. Ören, and J. Filipe, 35–48. Cham: Springer International Publishing.

Gratiet, L. L., and C. Cannamela. 2015. "Cokriging-Based Sequential Design Strategies Using Fast Cross-Validation Techniques for Multi-Fidelity Computer Codes". *Technometrics* 57:418–427.

Jeong, H.-D. J., J.-S. R. Lee, D. McNickle, and K. Pawlikowski. 2005. "Distributed Steady-State Simulation of Telecommunication Networks With Self-Similar Teletraffic". *Simulation Modeling Practice and Theory* 13:233–256.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13:455–492.

Kelton, W. D. 1986. "Replication Splitting and Variance for Simulating Discrete-Parameter Stochastic Processes". *Operations Research Letters* 4:275–279.

Kennedy, M. C., and A. O'Hagan. 2000. "Predicting the Output From a Complex Computer Code When Fast Approximations Are Available". *Biometrika* 87:1–13.

Nelson, B. L. 2011. "Thirty Years of 'Batch Size Effects"'. In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 393–400. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Pawlikowski, K. 1990. "Steady-State Simulation of Queueing Processes: A Survey of Problems and Solutions". *ACM Computing Surveys* 22:123–170.

Qian, P. Z. G., and C. F. J. Wu. 2008. "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments". *Technometrics* 50:192–204.

Rasmussen, C. E., and C. K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge: MIT.

Santner, T. J., B. J. Williams, and W. I. Notz. 2003. *The Design and Analysis of Computer Experiments*. New York: Springer.

Song, W. T. 1996. "On the Estimation of Optimal Batch Sizes in the Analysis of Simulation Output". *European Journal of Operational Research* 88 (2): 304–319.

Song, W. T., and B. W. Schmeiser. 1995. "Optimal Mean-Squared-Error Batch Sizes". *Management Science* 41:110–123.

Whitt, W. 1989. "Planning Queueing Simulations". *Management Science* 35:1341–1366.

Whitt, W. 1991. "The Efficiency of One Long Run Versus Independent Replications in Steady-State Simulation". *Management Science* 37:645–666.

Whitt, W. 2006. "Analysis for Design". In *Simulation*, edited by S. G. Henderson and B. L. Nelson, Handbooks in Operations Research and Management Science Vol. 13, 381–413. Elsevier.

Yang, F., B. E. Ankenman, and B. L. Nelson. 2008. "Estimating Cycle Time Percentile Curves for Manufacutring Systems via Simulation". *INFORMS Journal on Computing* 20:628–643.

AUTHOR BIOGRAPHIES

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 and her web page is https://sites.google.com/vt.edu/xi-chen-ise/home.

SAHAR HEMMATI is a Graduate Student in the Department of Industrial and Management Systems Engineering at West Virginia University. Her research interests include stochastic processes modeling, applied statistics and data analysis. Her email address is s.hemmati92@gmail.com.

FENG YANG is an Associate Professor in the Department of Industrial and Management Systems Engineering at West Virginia University. Her research interests include applied statistical methods, design of experiment methodology and application, and computer simulation of stochastic systems. Her email address and web page are, respectively, Feng.Yang@mail.wvu.edu and https://web.statler.wvu.edu/~yang/.