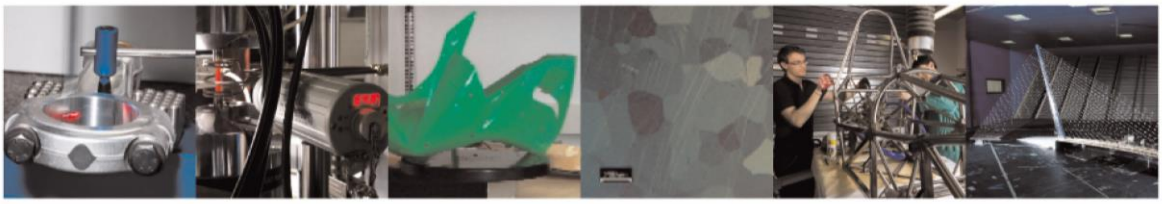
POLITECNICO
MILANO 1863

DIPARTIMENTO DI MECCANICA

mecc



Combining simulation experiments and analytical models with area-based accuracy for performance evaluation of manufacturing systems

Lin, Ziwei; Matta, Andrea; Shanthikumar, J. George

This is an Accepted Manuscript of an article published by Taylor & Francis in IISE TRANSACTIONS on 22 Feb 2019, available online:
https://doi.org/10.1080/24725854.2018.1490046

This content is provided under CC BY-NC-ND 4.0 license

BY NC ND

DIPARTIMENTO DI MECCANICA ■ POLITECNICO DI MILANO
via G. La Masa, 1 ■ 20156 Milano ■ EMAIL (PEC): pecmecc@cert.polimi.it
http://www.mecc.polimi.it
Rev. 0

Combining Simulation Experiments and Analytical Models with Different Area-Based Accuracy for Performance Evaluation of Manufacturing Systems

Ziwei Lin[1], Andrea Matta[2], and J. George Shanthikumar[3]

[1]Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University

[2] Department of Mechanical Engineering, Politecnico di Milano

[3] Krannert School of Management, Purdue University

Abstract

Simulation is considered as the most practical tool to estimate manufacturing system performance, but it is slow in its execution. Analytical models are generally available to provide fast but biased estimates of the system performance. These two approaches are commonly used distinctly in a sequential approach, or one as alternative to the other, for assessing manufacturing system performance. This paper proposes a method to combine simulation experiments with analytical results in the same performance evaluation model. The method is based on kernel regression and allows considering more than one analytical methods. High-fidelity model is combined with low-fidelity models for manufacturing system performance evaluation. Multiple area-based low-fidelity models can be considered for the prediction. The numerical results show that the proposed method is able to identify the reliability of low-fidelity models in different areas and provide higher accurate estimates. Comparison with alternative approaches shows the method is more accurate in a studied manufacturing application.

keyworks: performance evaluation, simulation, analytical modeling, multi-fidelity regression, manufacturing systems.

1 Introduction

1.1 Motivation

Analytical methods and simulation are two classical and effective tools for system performance evaluation in manufacturing system design. Queuing theory and Markov chains are the most utilized frameworks under which analytical methods are developed to rapidly estimate systems performance (Askin and Standridge, 1993; Buzacott and Shanthikumar, 1993; Gershwin, 1994;

Papadopoulos et al., 2009; Li and Meerkov, 2009; Tempelmeier and Kuhn, 1993). Analytical methods are fast and cheap. Their advantage is counterbalanced by the bias introduced to make analytical methods solvable for complex systems, either in the system assumptions (e.g., infinite buffer capacity) or in the mathematical derivation of the solution equations (e.g., memoryless property). In most situations, the more complex the system is, the larger the approximation introduced in the analytical method. Despite this disadvantage, analytical methods can be used to establish the general frameworks of complex systems and make rapid optimization possible.

Simulation is capable of providing highly accurate estimate of the expected system performance, except for the bias introduced by finite sample size in simulation runs and the detail level of the conceptual model. Nevertheless, simulation is usually expensive and requires a great deal of time in executing experiments. Generally, the higher the detail level of the simulation model, the more expensive the simulation project. To keep low the computational effort, regression and interpolating techniques are used to build surrogate models from simulation experiments (Kernel Regression (Wand and Jones, 1995), Kriging (Sacks et al., 1989), Neural Network (Haykin, 2009), Splines (Wahba, 1990), Support Vector Regression (Drucker et al., 1996)). Surrogate models provide a way to rapidly predict the system performance at unobserved points. However, they usually require a large amount of design points to perform well when the inherent response surface is not smooth enough or the dimension of the input is high.

Combining the properties of analytical methods and simulation for the system performance evaluation of manufacturing systems is the subject of this paper. Specifically, the paper deals with the cases in which multiple area-based analytical methods are available for rapid but rough estimation, by studying how data, collected from different experiments and having different accuracy, can be merged to improve the estimates of manufacturing system performance.

1.2 Problem

Low-fidelity models, in this paper low detail level simulation models or analytical methods, provide fast but approximate estimates while high-fidelity models (high detail level simulation models or system field data) provide accurate but slow estimates. Multi-fidelity regression modeling can use all the experimental data from both high-fidelity models and low-fidelity models to combine the advantages of both in order to build a more accurate and fast predictor. We assume that high-fidelity model generates accurate measurements of the system performance, whereas low-fidelity models capture some basic structural properties of the system response, such as the throughput of a manufacturing system increases when processing times decrease. Compared to single fidelity surrogate models (in which only high-fidelity data are used), the accuracy of multi-fidelity predictors can be improved thanks to the help of the basic features captured by low-fidelity models.

Moreover, when evaluating systems' performance, alternative low-fidelity models can be used. We assume that there is no hierarchy among these low-fidelity models for the prediction, which means it is difficult to know in advance which low-fidelity models are better than others. This is to model the fact that most of low-fidelity models applied in manufacturing systems usually have area-based accuracy in estimating the system performance. They could perform well in a specific area of the domain while roughly in others. For example, an analytical method developed under the assumption of infinite buffer space would provide accurate estimates when the input buffer size is large enough, but have poor behavior when limited buffer space is required and blockage phenomena happens. On the contrary, an analytical method considering no buffer space would behave inversely. Despite this, both analytical methods could model well the monotonicity of production rate as a function of service rates. Therefore, considering different low-fidelity models in different areas is important in the system performance evaluation to make full use of the information provided by low-fidelity models and make the estimates more accurate.

The problem investigated in this paper is to develop a surrogate model that combines experimental data from high-fidelity model with outputs of low-fidelity models for performance evaluation of manufacturing systems. The developed model is expected to be more accurate than single fidelity surrogate models with the help of low-fidelity models. Negligible additional time is needed thanks to the fast property of low-fidelity models. Furthermore, the developed predictor is expected to select helpful information automatically from multiple non-hierarchical low-fidelity models to judge which model is better than others, and in which area.

1.3 Related literature

A large number of researches about multi-fidelity modeling methods have been proposed in literature. Among these methods, Gaussian Process based methods transform both high-fidelity data and low-fidelity data into a realization of one set of correlated Gaussian Processes variables and estimate the new variables through the conditional distribution property of multivariate normal distribution (i.e., best linear unbiased predictor). Co-Kriging (Cressie, 1992) and Alternative Co-Kriging (Han et al., 2012), which are applied to fit aerodynamic data, belong to this class. More than one low-fidelity model can be considered by extending these two techniques (Kennedy and O'Hagan, 2000; Yamazaki and Mavriplis, 2013).

Scaling function based methods correct the outputs of low-fidelity model $y_l(\boldsymbol{x})$ through scaling functions with the responses of high-fidelity model as the scale baseline:

$$\hat{y}_h(\boldsymbol{x}) = \hat{\phi}(\boldsymbol{x}) y_l(\boldsymbol{x}) + \hat{\gamma}(\boldsymbol{x}),$$

where the multiplicative scaling functions $\hat{\phi}(\boldsymbol{x})$ and additive scaling functions $\hat{\gamma}(\boldsymbol{x})$ can be estimated using regression or interpolating techniques. Most of multi-fidelity meta-models belong to this class. Among these, multiplicative scaling function is used by Haftka (1991) and estimated by first order Taylor series. First order Taylor series are also used in Chang et al. (1993) for aircraft design. Other fitting techniques can be also used to estimate the scaling functions, like Neural Network (Watson and Gupta (1996), for modeling microstrip vias), Kriging (Huang et al. (2006), for die wear minimization; Gano et al. (2005), for airfoil design), Radial Basis Function (Sun et al. (2010), for honeycomb-type cellular materials), Least Squares method (Sun et al. (2011), for sheet metal forming process; Osorio and Bierlaire (2013), for urban transportation problems), Support Vector Regression (Zhou et al. (2015), for long cylinder pressure vessel design). Hierarchical Kriging (Han and Görtz, 2012) and Improved Hierarchical Kriging (Hu et al., 2016), which are applied to model airfoils, estimate the additive and multiplicative scaling function by Kriging and Maximum Likelihood method, respectively. Han et al. (2013) iteratively update the multiplicative scaling function as a polynomial and estimate the additive scaling function by Kriging for modeling airfoil data.

Space Mapping is another class of multi-fidelity meta modeling. It assumes that the input space of low-fidelity model is different from the input space of high-fidelity model and constructs a transformation $\boldsymbol{P}$ from the high-fidelity model parameters space to the low-fidelity model parameters space such that

$$y_h(\boldsymbol{x}) \approx y_l(\boldsymbol{P}(\boldsymbol{x})).$$

For instance, Bandler et al. (1994, 1995) apply space mapping technique in electromagnetic optimization problem, Bandler et al. (2001) in microstrip problem, Bakr et al. (2000) in microwave circuits optimization problem. Robinson et al. (2008) correct the space mapping technique using a quadratic Taylor series expansion in wing design problem and flapping-flight problem.

In addition to these three classes of multi-fidelity modeling techniques, many other methods have been proposed in literature. For example, Knowledge Based Neural Network (KBNN) (Wang and Zhang, 1997), which is proposed for microwave design, adds a knowledge layer in a neural network to consider low-fidelity models in the prediction. Leary et al. (2003) use simplified Knowledge Based Neural Network for beam design problem. Chen et al. (2015) and Xu et al. (2016) evaluate and optimize the system performance of a job shop using low-fidelity outputs to sort the alternative solutions, this helps making the fitting function smoother before the prediction.

Summarizing, several methods have been proposed in literature to use data from different experiments. Most of them only consider a single low-fidelity model in the modeling, or they require prior assumptions about the relative importance of low-fidelity models in different areas of the explored region. For example, Yamazaki and Mavriplis (2013) assume the importance of low-fidelity models remain the same through the whole domain, Hierarchical Kriging (Han and Görtz, 2012) and Kennedy and O'Hagan (2000) define a hierarchy among the low-fidelity models, Hu et al. (2016) pre-define a quadratic shape of the importance level function before the prediction. Knowledge Based Neural Network (Wang and Zhang, 1997) method considers different low-fidelity models in different regions with additional effort used to decide the number and the shape of the boundaries as well as the number of regions before the prediction.

1.4 Contribution

The paper proposes a method to integrate experimental data from high-fidelity and low-fidelity models for the estimation of manufacturing systems performance. The low-fidelity models provide the global trend of the fitting function while the high-fidelity model provides accurate estimates used for correcting the low-fidelity outputs. Multiple non-hierarchical low-fidelity models are considered in the method. More specifically, these low-fidelity models have variable reliability in estimating system performance in different areas. The developed method chooses helpful low-fidelity models in different areas automatically and driven by data, i.e., according to the information collected from the experiments.

The research contribution of this paper is twofold:

1. A method to combine high-fidelity model with low-fidelity models is proposed to evaluate the performance of manufacturing systems. Most of proposed literature about multi-fidelity regression techniques focus on performance evaluation and optimization in engineering design

problems, microwave, aircraft, etc. However, very few papers deal with manufacturing systems. Chen et al. (2015) and Xu et al. (2016) analysis job shops combining queuing with simulation, Lin et al. (2016) combine Continuous Time Markov Chain method with simulation for analyzing closed-loop flexible assembly system, but they only consider a single low-fidelity model in the prediction. This work will help towards the construction of a more general framework, in which analytical approaches and simulation coexist in the same model, to analyze and design manufacturing systems. Currently, they are commonly used sequentially (first analytical models and then detailed simulation) or as one alternative of the other.

2. A data-driven method is developed to select helpful information from multiple area-based low-fidelity models. No hierarchy is defined among these low-fidelity models and no predefined shape of low-fidelity models' importance level function is required. Neither the shapes of the boundaries nor the number of the boundaries and area are required to be decided before the prediction. The weights assigned to low-fidelity models are calculated automatically from experimental data.

The proposed method is tested numerically in a single-dimensional case and applied to a 4-dimensional closed-loop flexible assembly system. As shown in the numerical results, the method is capable of judging the reliability of low-fidelity models in different areas and combines them efficiently during the prediction. Numerical results will also show that using multiple low-fidelity models allow to reach better accuracy compared to using a single low-fidelity model.

1.5 Outline

This paper is organized as follows. Section 2 presents the notation throughout the paper and describes the problem in a formal way. Section 3 describes the proposed method and the procedure to execute it. Section 4 applies the proposed method to a case designed for testing purpose. Section 5 presents a manufacturing application of the proposed method. Finally, conclusion and guidelines for future developments are drawn in section 6.

2 Notation

Given a system we want to analyze, we assume a high-fidelity model is capable of providing highly accurate estimate of the system performance (stochastic or deterministic). Besides, m low-fidelity models are available for this system but their predictions are approximate. Each of them is considered to embody some specific features of the given system and may be reliable in different areas or under some conditions. The low-fidelity outputs are considered as deterministic (analytical method outputs or low detail simulation outputs with very tight confidence interval). The notation of the system is listed below.

$\mathcal{D} \subset \mathbb{R}^d$	: domain of possible system configurations
d	: dimension of system configuration
$\boldsymbol{x} = [x_1, \cdots, x_d]^T \in \mathcal{D}$	: system configuration
$\mathcal{J} = \{1, \cdots, m\}$	: set of low-fidelity models
$\boldsymbol{y}_l(\boldsymbol{x}) = [y_{l_1}(\boldsymbol{x}), \cdots, y_{l_m}(\boldsymbol{x})]^T$	: outputs of low-fidelity models at the point $\boldsymbol{x}$
$y_h(\boldsymbol{x})$	: response of high-fidelity model at the point $\boldsymbol{x}$

The system performance estimates are denoted with "l" and "h" according to convention. "l" means these estimates come from low-fidelity models, while those denoted with "h" come from high-fidelity model.

We assume that a Design of Experiment (DOE) containing n design points has been developed and executed. Two kinds of estimates from the experiments are available. One is the response of the high-fidelity model, the other is the output of the m different low-fidelity models. For the sake of convenience, the notations of all the DOE data are listed below. The index "0" indicates that these data belong to the initial design.

$\mathcal{N} = \{1, \cdots, n\}$	: set of DOE points
$\boldsymbol{x}_i^0 = [x_{i1}^0, \cdots, x_{id}^0]^T, i \in \mathcal{N}$	: system configuration of design point i
$\boldsymbol{X}^0 = [\boldsymbol{x}_1^0, \cdots, \boldsymbol{x}_n^0]^T$	: $n \times d$ matrix of system configurations in the design
$\boldsymbol{Y}_l^0 = [\boldsymbol{y}_l(\boldsymbol{x}_1^0), \cdots, \boldsymbol{y}_l(\boldsymbol{x}_n^0)]^T$	: $n \times m$ matrix of low-fidelity models outputs
$\boldsymbol{Y}_h^0 = [y_h(\boldsymbol{x}_1^0), \cdots, y_h(\boldsymbol{x}_n^0)]^T$	: $n \times 1$ vector of high-fidelity model responses

We are interested in estimating the expected system performance at the unknown point $\boldsymbol{x}$ with the information collected from the DOE (i.e., $\boldsymbol{X}^0, \boldsymbol{Y}_l^0, \boldsymbol{Y}_h^0$) and the outputs of low-fidelity models at this unknown point (i.e., $\boldsymbol{y}_l(\boldsymbol{x})$):

$$\widehat{Ey_h}(\boldsymbol{x}) = \hat{y}(\boldsymbol{x} | \boldsymbol{X}^0, \boldsymbol{Y}_l^0, \boldsymbol{Y}_h^0, \boldsymbol{y}_l(\boldsymbol{x})).$$

3 Extended Kernel Regression

3.1 Predictor proposed

The main idea of Extended Kernel Regression (EKR) method is correcting low-fidelity outputs through several high-fidelity responses, locally regressing the corrected data provided by the same low-fidelity model and combining multiple low-fidelity models according to their local prediction errors. Since each design point contributes to the estimation, there will be no waste of data in the predictor.

3.1.1 Corrected low-fidelity surrogate models

The knowledge about some structural properties of the system performance function embedded in low-fidelity models can help to build the estimator, particularly when the unobserved point is far from the initial design points and the spatial correlation function is not likely to hold. The approximate outputs of low-fidelity models can be corrected by considering the responses of high-fidelity model as the benchmarks. Additive scaling function (Lewis and Nash, 2000) assumes that a certain low-fidelity model has similar prediction error (i.e., has similar deviation between high-fidelity response and low-fidelity output itself) for points that are close to each other. Based on this assumption, the output of each low-fidelity model at the unobserved point (i.e., $y_{l_j}(\boldsymbol{x}), \forall j \in \mathcal{J}$) can be modified as follows:

$$\tilde{y}_i^{l_j}(\boldsymbol{x}) = y_{l_j}(\boldsymbol{x}) + (y_h(\boldsymbol{x}_i^0) - y_{l_j}(\boldsymbol{x}_i^0)), \forall i \in \mathcal{N}, \forall j \in \mathcal{J}, \tag{1}$$

where $y_{l_j}(\boldsymbol{x})$ is the output of the j-th low-fidelity model at the unobserved point $\boldsymbol{x}$. The difference $y_h(\boldsymbol{x}_i^0) - y_{l_j}(\boldsymbol{x}_i^0)$ is the deviation between high-fidelity model and the j-th low-fidelity model (i.e., the additive scaling function) evaluated at the i-th initial design points $\boldsymbol{x}_i^0$. Each output of the m low-fidelity models at the unobserved point is modified through each of the n design points. A total $n \times m$ corrected low-fidelity outputs are obtained in this step.

Other scaling functions can be used for the correction as well. For example, multiplicative scaling function (Haftka, 1991), which assumes that the ratio of high-fidelity model response over a certain low-fidelity model output is smooth within local area, is also a reasonable correction method. The corrected outputs provided by low-fidelity models and modified by multiplicative scaling function have the form:

$$\tilde{y}_i^{l_j}(\boldsymbol{x}) = \frac{y_h(\boldsymbol{x}_i^0)}{y_{l_j}(\boldsymbol{x}_i^0)} \cdot y_{l_j}(\boldsymbol{x}), \forall i \in \mathcal{N}, \forall j \in \mathcal{J}, \tag{2}$$

where $\frac{y_h(\boldsymbol{x}_i^0)}{y_{l_j}(\boldsymbol{x}_i^0)}$ is the ratio of high-fidelity model response over the j-th low-fidelity model output (i.e., the multiplicative scaling function) that is evaluated at the i-th initial design points $\boldsymbol{x}_i^0$. Different scaling functions can be used for different low-fidelity models in the same EKR model.

These corrected outputs are expected to have more reliable prediction performance if their scaling functions are estimated by the initial design points close to the unobserved point. Therefore, local polynomial regression (Wand and Jones, 1995) is used to locally fit these data with distance-based weights. For the corrected outputs provided by the same low-fidelity model, a local polynomial is fitted around the unobserved point $\boldsymbol{x}$:

$$\begin{aligned}
P_{\boldsymbol{x}}^{l_j}(\boldsymbol{u}; \boldsymbol{a}_{\boldsymbol{x}}^{l_j}) = a_0^{l_j} &+ a_{11}^{l_j}(u_1 - x_1) + \cdots + a_{1d}^{l_j}(u_d - x_d) \\
&+ a_{21}^{l_j}(u_1 - x_1)^2 + \cdots + a_{2d}^{l_j}(u_d - x_d)^2 \\
&+ \cdots \\
&+ a_{p1}^{l_j}(u_1 - x_1)^p + \cdots + a_{pd}^{l_j}(u_d - x_d)^p, \forall j \in \mathcal{J},
\end{aligned}$$

where p is the degree of the polynomial, $\boldsymbol{a}_{\boldsymbol{x}}^{l_j} = [a_0^{l_j}, a_{11}^{l_j}, \cdots, a_{1d}^{l_j}, \cdots, a_{p1}^{l_j}, \cdots, a_{pd}^{l_j}]^T$ is an unknown parameter vector to be estimated, $\boldsymbol{x}$ is the unobserved point, $\boldsymbol{u}$ is the variable vector of the polynomial and $x_i, u_i, (i = 1, \cdots, d)$ are components of $\boldsymbol{x}, \boldsymbol{u}$, respectively. The unknown parameter vector $\boldsymbol{a}_{\boldsymbol{x}}^{l_j}$ is estimated by minimizing the *weighted square error* between the corrected outputs $\tilde{y}_i^{l_j}(\boldsymbol{x}), \forall i \in \mathcal{N}$ and the value of the fitted polynomial at each design point $P_{\boldsymbol{x}}^{l_j}(\boldsymbol{x}_i^0; \boldsymbol{a}_{\boldsymbol{x}}^{l_j}), \forall i \in \mathcal{N}$:

$$WSE_{l_j}(\boldsymbol{x}) = \sum_{i \in \mathcal{N}} \frac{K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})}{\sum_{i \in \mathcal{N}} K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})} (\tilde{y}_i^{l_j}(\boldsymbol{x}) - P_{\boldsymbol{x}}^{l_j}(\boldsymbol{x}_i^0; \boldsymbol{a}_{\boldsymbol{x}}^{l_j}))^2, \forall j \in \mathcal{J}. \tag{3}$$

The weights for each initial design point are calculated through Gaussian Kernel function:

$$K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x}) = \prod_{k=1}^{d} \exp\left\{-\frac{1}{2\theta_{1k}}(x_{ik}^0 - x_k)^2\right\}, \forall i \in \mathcal{N}, \tag{4}$$

where $\boldsymbol{\Theta}_1 = \text{diag}\{\theta_{11}, \cdots, \theta_{1d}\}$, and $\theta_{1k} > 0, k = 1, \cdots, d$ are parameters selected according to some specific criteria. These parameters control the influence range of the design points on the prediction. Gaussian Kernel is a function of the distance between the design point and the unobserved point. It gives large weights to the design points close to the unobserved point and small weights to the design points far away. Therefore, it allows each design point to have an importance related to the unobserved point $\boldsymbol{x}$. The closer the design points, the larger their importance. For the convenience of derivation, equation (3) can be written in matrix form:

$$WSE_{l_j}(\boldsymbol{x}) = (\mathrm{tr}(\boldsymbol{W_x}))^{-1}(\tilde{\boldsymbol{Y}}_{l_j} - \boldsymbol{X_x}\boldsymbol{a}_{\boldsymbol{x}}^{l_j})^T\boldsymbol{W_x}(\tilde{\boldsymbol{Y}}_{l_j} - \boldsymbol{X_x}\boldsymbol{a}_{\boldsymbol{x}}^{l_j}), \forall j \in \mathcal{J}, \tag{5}$$

where

$$\boldsymbol{X_x} = \begin{bmatrix} 1 & (\boldsymbol{x}_1^0 - \boldsymbol{x})^T & \cdots & \left[(\boldsymbol{x}_1^0 - \boldsymbol{x})^p\right]^T \\ 1 & (\boldsymbol{x}_2^0 - \boldsymbol{x})^T & \cdots & \left[(\boldsymbol{x}_2^0 - \boldsymbol{x})^p\right]^T \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (\boldsymbol{x}_n^0 - \boldsymbol{x})^T & \cdots & \left[(\boldsymbol{x}_n^0 - \boldsymbol{x})^p\right]^T \end{bmatrix}$$

is an $n \times (dp+1)$ matrix. $\boldsymbol{W_x} = \mathrm{diag}\{K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_1^0 - \boldsymbol{x}), \cdots, K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_n^0 - \boldsymbol{x})\}$ is an $n \times n$ diagonal matrix containing the kernel values of corresponding initial design points, $\tilde{\boldsymbol{Y}}_{l_j} = [\tilde{y}_1^{l_j}(\boldsymbol{x}), \cdots, \tilde{y}_n^{l_j}(\boldsymbol{x})]^T$ and $\mathrm{tr}(\boldsymbol{W_x})$ is the trace of $\boldsymbol{W_x}$. By setting the partial derivative of equation (5) equal to zero:

$$\frac{\partial WSE_{l_j}(\boldsymbol{x})}{\partial \boldsymbol{a}_{\boldsymbol{x}}^{l_j}} = -2(\mathrm{tr}(\boldsymbol{W_x}))^{-1}\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W_x}(\tilde{\boldsymbol{Y}}_{l_j} - \boldsymbol{X_x}\boldsymbol{a}_{\boldsymbol{x}}^{l_j}) = \boldsymbol{0},$$

we can derive the estimate of $\boldsymbol{a}_{\boldsymbol{x}}^{l_j}$, which has the following closed form:

$$\hat{\boldsymbol{a}}_{\boldsymbol{x}}^{l_j} = (\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W_x}\boldsymbol{X_x})^{-1}\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W_x}\tilde{\boldsymbol{Y}}_{l_j}, \forall j \in \mathcal{J}. \tag{6}$$

The estimate of the system performance at the unobserved point $\boldsymbol{x}$ provided by the j-th corrected low-fidelity surrogate model is obtained by plugging the unknown point $\boldsymbol{x}$ into the fitted polynomial: $\hat{y}_{l_j}(\boldsymbol{x}) = P_{\boldsymbol{x}}^{l_j}(\boldsymbol{x}; \hat{\boldsymbol{a}}_{\boldsymbol{x}}^{l_j}) = \hat{a}_0^{l_j}$, which has the form:

$$\hat{y}_{l_j}(\boldsymbol{x}) = \boldsymbol{e}_1^T\hat{\boldsymbol{a}}_{\boldsymbol{x}}^{l_j} = \boldsymbol{e}_1^T(\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W_x}\boldsymbol{X_x})^{-1}\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W_x}\tilde{\boldsymbol{Y}}_{l_j}, \forall j \in \mathcal{J}, \tag{7}$$

where $\boldsymbol{e}_1$ is a $(dp+1)$-dimensional vector whose first element is 1 and the rest are 0.

If we set the polynomial degree $p = 0$ (i.e., $\boldsymbol{X_x}$ is an n-dimensional vector whose elements are all equal to 1), the predictor in equation (7) turns to the Nadaraya-Watson estimator (Nadaraya, 1964; Watson, 1964), which regresses the data by weighted averaging:

$$\hat{y}_{l_j}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \frac{K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})}{\sum_{i\in\mathcal{N}} K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})} \tilde{y}_i^{l_j}(\boldsymbol{x}), \forall j \in \mathcal{J}. \tag{8}$$

Nadaraya-Watson estimator is easy and quick in calculating, but has larger bias in the boundary (Wand and Jones, 1995). According to Wand and Jones (1995), higher degree p can improve the asymptotic performance of the predictor, i.e., reduce the prediction bias. However, predictor with higher degree p has larger variance and large sample size might be required for substantial improvement in practical application, especially beyond cubic fits. In addition, odd degree fits have attractive boundary properties. Therefore, $p = 1$ and $p = 3$ are recommended by Wand and Jones (1995).

3.1.2 Combination of low-fidelity models

As described in section 1, low-fidelity models usually have area-based reliability in estimating system performance. In this work, reliability of low-fidelity model l_j, at prediction point $\boldsymbol{x}$, is defined as proportional to accuracy of its corresponding corrected surrogate model $\hat{y}_{l_j}(\boldsymbol{x})$. Given an unobserved point to be evaluated, it is very important to find out which low-fidelity models can provide more reliable information for the prediction in this local area. Then, higher weights can be assigned to them when combining all the estimates provided by corrected low-fidelity surrogate models (i.e., $\hat{y}_{l_j}(\boldsymbol{x}), \forall j \in \mathcal{J}$ in equation (7)):

$$\hat{y}(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \frac{\text{Reliability of } l_j \text{ at } \boldsymbol{x}}{\text{Sum of all the reliability at } \boldsymbol{x}} \hat{y}_{l_j}(\boldsymbol{x}).$$

A reasonable index of the local reliability of low-fidelity models is the estimated weighted square error obtained by substituting equation (6) into equation (5):

$$W\hat{S}E_{l_j}(\boldsymbol{x}) = (\text{tr}(\boldsymbol{W_x}))^{-1} \tilde{\boldsymbol{Y}}_{l_j}^T (\boldsymbol{W_x} - \boldsymbol{W_x}^T \boldsymbol{X_x} (\boldsymbol{X_x}^T \boldsymbol{W_x} \boldsymbol{X_x})^{-1} \boldsymbol{X_x}^T \boldsymbol{W_x}) \tilde{\boldsymbol{Y}}_{l_j}, \forall j \in \mathcal{J}. \tag{9}$$

Since the outputs of low-fidelity models are corrected before the regression, the local polynomial regression predictor in equation (7) is in fact regressing the scaling function (e.g., the bias or the ratio) of each low-fidelity model. Hence, the estimated weighted square error in equation (9) indicates how well the scaling function of a certain low-fidelity model can be locally fitted by polynomial. The estimated weighted square error can be considered as the estimate of the square of the local prediction error. The lower the estimated weighted square error, the more reliable the low-fidelity model in this local area for the prediction.

If we set $p = 0$ (i.e., Nadaraya-Watson estimator case), the $W\hat{S}E_{l_j}(\boldsymbol{x})$ has the following form:

$$W\hat{S}E_{l_j}(\boldsymbol{x}) = \sum_{i \in \mathcal{N}} \frac{K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})}{\sum_{i \in \mathcal{N}} K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})} (\tilde{y}_i^{l_j}(\boldsymbol{x}) - \hat{y}_{l_j}(\boldsymbol{x}))^2, \forall j \in \mathcal{J}, \tag{10}$$

where $\hat{y}_{l_j}(\boldsymbol{x})$ is the weighted average of $\tilde{y}_i^{l_j}(\boldsymbol{x}), \forall i \in \mathcal{N}$ as shown in equation (8). According to the form of equation (10), $W\hat{S}E_{l_j}(\boldsymbol{x})$ can be regarded as the weighted variance of $\tilde{y}_i^{l_j}(\boldsymbol{x}), \forall i \in \mathcal{N}$. Therefore, $W\hat{S}E_{l_j}(\boldsymbol{x})$ indicates the variability of $\tilde{y}_i^{l_j}(\boldsymbol{x}), \forall i \in \mathcal{N}$ in this local area. The smaller the weighted variance, the more stable these data in this local area. This inference is consistent with common sense (i.e., the smoother the data, the more accurate the estimate) and justifies the use of $W\hat{S}E_{l_j}$.

The final predictor is obtained by weighted averaging all the system performance estimates provided by corrected low-fidelity surrogate models (i.e., $\hat{y}_{l_j}(\boldsymbol{x}), \forall j \in \mathcal{J}$ in equation (7)) with weights related to the square of the local prediction error (i.e., $W\hat{S}E_{l_j}(\boldsymbol{x}), \forall j \in \mathcal{J}$ in equation (9)):

$$\hat{y}_{EKR}(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} w_{l_j}(\boldsymbol{x}) \hat{y}_{l_j}(\boldsymbol{x}), \tag{11}$$

where

$$w_{l_j}(\boldsymbol{x}) = \frac{K_{2,\theta_2}(W\hat{S}E_{l_j}(\boldsymbol{x}))}{\sum_{j \in \mathcal{J}} K_{2,\theta_2}(W\hat{S}E_{l_j}(\boldsymbol{x}))}, \tag{12}$$

and $K_{2,\theta_2}(\cdot)$ has the following form:

$$K_{2,\theta_2}(W\hat{S}E_{l_j}(\boldsymbol{x})) = \exp\left\{ -\frac{W\hat{S}E_{l_j}(\boldsymbol{x})}{2\theta_2 WSE_{min}(\boldsymbol{x})} \right\}, \forall j \in \mathcal{J}, \tag{13}$$

$$WSE_{min}(\boldsymbol{x}) = \min_{j \in \mathcal{J}} \{W\hat{S}E_{l_j}(\boldsymbol{x})\}.$$

θ_2 is an unknown parameter to be selected according to some criterion. It controls the decline rate of the K_{2,θ_2} value as the error increases. After selecting θ_2 value, a fixed K_{2,θ_2} value is assigned to the low-fidelity model with the smallest local prediction error while smaller values are assigned to other low-fidelity models. The larger the error, the smaller the weight. The use of $WSE_{min}(\boldsymbol{x})$ can be regarded as normalizing $W\hat{S}E_{l_j}(\boldsymbol{x}), \forall j \in \mathcal{J}$ according to the performance of the local best low-fidelity model and combining low-fidelity models according to the ratios of the local prediction errors rather than the errors themselves. It can mitigate the influence of the errors' magnitudes (the magnitudes of the errors might be different in different areas), at the same time can reduce the effect of the existence of extreme large local prediction errors.

The EKR predictor in equation (11) combines low-fidelity models according to their local prediction errors. Higher weights are assigned to low-fidelity models that have better local behaviors and vice versa. The calculation of local prediction errors is data-driven. No pre-knowledge is required, which makes the EKR predictor be capable of judging the reliability of low-fidelity models in different areas during the prediction.

In case of linear fitting, i.e. $p = 1$, a proof of convergence of the predictor as $n \to \infty$ is presented in Appendix B. Furthermore, an approximate prediction interval with confidence level $1 - \alpha$ is provided for the EKR predictor under mild assumptions:

$$y_h(\boldsymbol{x}) \in [\hat{y}_{EKR}(\boldsymbol{x}) - Z_{\alpha/2}\hat{s}(\boldsymbol{x}), \hat{y}_{EKR}(\boldsymbol{x}) + Z_{\alpha/2}\hat{s}(\boldsymbol{x})]$$

where

$$\hat{s}(\boldsymbol{x})^2 = W\hat{S}E(\boldsymbol{x})\left(1 + \frac{1}{2^{d/2}\mathrm{tr}(\boldsymbol{W_x})}\right),$$

$$W\hat{S}E(\boldsymbol{x}) = (\mathrm{tr}(\boldsymbol{W_x}))^{-1}\tilde{\boldsymbol{Y}}^T(\boldsymbol{W_x} - \boldsymbol{W_x}^T\boldsymbol{X_x}(\boldsymbol{X_x}^T\boldsymbol{W_x}\boldsymbol{X_x})^{-1}\boldsymbol{X_x}^T\boldsymbol{W_x})\tilde{\boldsymbol{Y}},$$

$\tilde{\boldsymbol{Y}} = \left[\sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})\tilde{y}_1^{l_j}(\boldsymbol{x}), \cdots, \sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})\tilde{y}_n^{l_j}(\boldsymbol{x})\right]^T$ and $Z_{\alpha/2}$ is the quantile value of the standard normal distribution. More details can be found in Appendix A.

3.2 Implementation details

3.2.1 Parameters estimation

In the EKR predictor described in section 3.1, model parameters (i.e., $\boldsymbol{\Theta}_1$ and θ_2) are unknown and require to be estimated before the prediction. It is reasonable to choose the model parameters that minimize the Mean Integral Square Error (MISE) of the EKR predictor at the whole domain:

$$MISE = \int_{\boldsymbol{x}\in\mathcal{D}} (y_h(\boldsymbol{x}) - \hat{y}_{EKR}(\boldsymbol{x}; \boldsymbol{\Theta}_1, \theta_2))^2 f(\boldsymbol{x})d\boldsymbol{x},$$

where $f(\boldsymbol{x})$ is the density function of $\boldsymbol{x}$. Nevertheless, the MISE depends on the unknown function of the system performance. Leave-one-out cross-validation score method (Wasserman,

2006) is used for the parameters' selection:

$$\{\hat{\mathbf{\Theta}}_1, \hat{\theta}_2\} = \arg \min_{\mathbf{\Theta}_1, \theta_2 > 0} \left\{ M\hat{I}SE = \frac{1}{n} \sum_{i \in \mathcal{N}} (y_h(\boldsymbol{x}_i^0) - \hat{y}_{EKR_{-i}}(\boldsymbol{x}_i^0; \mathbf{\Theta}_1, \theta_2))^2 \right\}, \tag{14}$$

where $\hat{y}_{EKR_{-i}}(\cdot; \mathbf{\Theta}_1, \theta_2)$ is the EKR predictor built from the DOE data with information related to the i-th design point removed. Therefore, the design point $\boldsymbol{x}_i^0$ is unobserved (i.e., no high-fidelity data is available) for this predictor. For each design point in the DOE, a different EKR predictor is built with the same procedure but different DOE data. The use of $\hat{y}_{EKR_{-i}}(\cdot; \mathbf{\Theta}_1, \theta_2)$ can avoid the collected data (i.e., the responses of high-fidelity model) being used twice: to build the EKR predictor and to estimate the MISE.

3.2.2 System configuration normalization

Normalizing the system configurations at design points and unobserved points (i.e., $\boldsymbol{x}_i^0, \forall i \in \mathcal{N}$ and $\boldsymbol{x}$) is recommended when implement the EKR method. It can mitigate the influence of the magnitudes of the system configuration's different dimensions. The system configurations can be normalized into $[0, 1]^d$ as follows:

1. Set the normalization parameters:

   $\boldsymbol{x}_{max} = [x_{max,1}, \cdots, x_{max,d}]^T$ where $x_{max,k} = \max_{i \in \mathcal{N}}\{x_{ik}^0\}, \forall k = 1, \cdots, d,$

   $\boldsymbol{x}_{min} = [x_{min,1}, \cdots, x_{min,d}]^T$ where $x_{min,k} = \min_{i \in \mathcal{N}}\{x_{ik}^0\}, \forall k = 1, \cdots, d.$

2. Normalize the system configuration of design points $\boldsymbol{x}_i^0$ and unobserved point $\boldsymbol{x}$:

   $x_{ik}^0 = (x_{ik}^0 - x_{min,k})/(x_{max,k} - x_{min,k}), \forall k = 1, \cdots, d, \forall i \in \mathcal{N},$

   $x_k = (x - x_{min,k})/(x_{max,k} - x_{min,k}), \forall k = 1, \cdots, d.$

3.2.3 Algorithm

Algorithm 1 describes in detail how to implement the EKR method. Firstly, design points are sampled and executed in Step 1. The information is collected by executing multiple low-fidelity models and one high-fidelity model at each sampled design point. Secondly, DOE data are normalized and kernel parameters $\mathbf{\Theta}_1, \theta_2$ are estimated as described in this section in Step2. Thirdly, low-fidelity outputs at the unobserved point are collected. Finally, the estimate of the expected system performance at the unobserved point is obtained using the EKR predictor built as described in section 3.1 in Step 3.

Algorithm 1 Procedure to implement the EKR method

```
Step 1 Develop and execute DOE:
$x_i^0 \in \mathcal{D}, i \in \mathcal{N} = \{1, \ldots, n\} \leftarrow$ system configurations of $n$ sampled design points
$y_{l_j}(\cdot), j \in \mathcal{J} = \{1, \cdots, m\} \leftarrow$ $m$ available low-fidelity models
$y_h(\cdot) \leftarrow$ a feasible high-fidelity model
for $i = 1, \ldots, n$ do
  Execute high-fidelity model at $\boldsymbol{x}_i^0$ and collect $y_h(\boldsymbol{x}_i^0)$
  for $j = 1, \ldots, m$ do
    Execute the $j$-th low-fidelity model at $\boldsymbol{x}_i^0$ and collect $y_{l_j}(\boldsymbol{x}_i^0)$
  end for
end for

Step 2 Estimate parameters:
$p \leftarrow$ degree of polynomial
Set $\boldsymbol{x}_{max}, \boldsymbol{x}_{min}$ and normalize $\boldsymbol{x}_i^0, i \in \mathcal{N}$
Estimate $\boldsymbol{\Theta}_1, \theta_2$ using equation (14)

Step 3 Evaluate unknown point:
$\boldsymbol{x} \in \mathcal{D} \leftarrow$ System configuration to be evaluated
for $j = 1, \ldots, m$ do
  Execute the $j$-th low-fidelity model at $\boldsymbol{x}$ and collect $y_{l_j}(\boldsymbol{x})$
end for
Normalize $\boldsymbol{x}$
for $i = 1, \ldots, n$ do
  Calculate $K_{1,\boldsymbol{\Theta}_1}(\boldsymbol{x}_i^0 - \boldsymbol{x})$ using equation (4)
end for
for $j = 1, \ldots, m$ do
  for $i = 1, \ldots, n$ do
    Calculate $\tilde{y}_i^{l_j}(\boldsymbol{x})$ using equation (1) or equation (2)
  end for
  Calculate $\hat{y}_{l_j}(\boldsymbol{x})$ using equation (7)
  Calculate $W\hat{S}E_{l_j}(\boldsymbol{x})$ using equation (9)
  Calculate $K_{2,\theta_2}(W\hat{S}E_{l_j}(\boldsymbol{x}))$ using equation (13)
end for
Calculate $\hat{y}_{EKR}(\boldsymbol{x})$ using equation (11)
```

4 Illustrative example

The proposed method is applied in a case specially designed for testing purpose and a better understanding of the method.

4.1 System description

As shown in Figure 1, one high-fidelity model and two low-fidelity models defined in $[0, 1]$ are considered in this case. The high-fidelity model is

$$y_h(x) = w(x)\sin(28x) + (1 - w(x))\sin(15x) + 10(x - 0.5)^2 + 4,$$

where $w(x)$ is a reversed sigmoid function:

$$w(x) = 1/(1 + \exp\{25(x - 0.5)\}).$$

The two low-fidelity models are:

$$y_{l_1}(x) = \sin(28x) + 2 \quad \text{and} \quad y_{l_2}(x) = \sin(15x) + 2.$$

The high-fidelity model is constructed by three components: low-fidelity model y_{l_1}, low-fidelity model y_{l_2} and a quadratic bias term. Low-fidelity model y_{l_1} is much more important than low-fidelity model y_{l_2} in $[0, 0.5)$, while less important in $(0.5, 1]$. All models are deterministic.

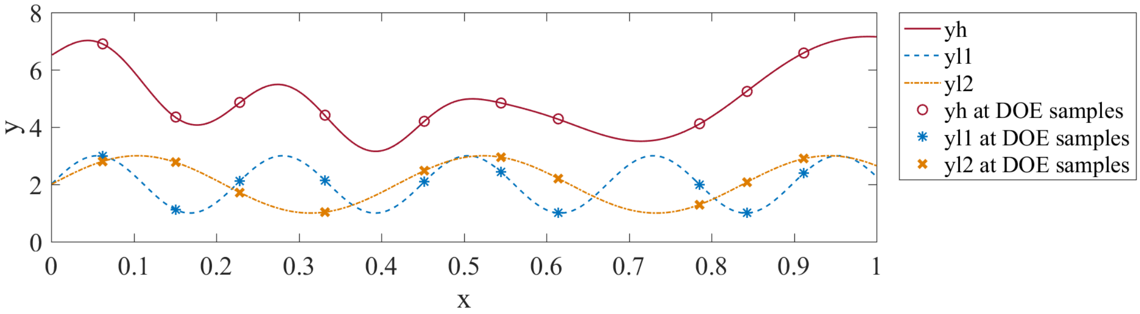


Figure 1: High-fidelity model and two different low-fidelity models in the test case as well as 10 sampled points in the DOE.

4.2 Algorithm implementation

An implementation of Algorithm 1 is presented in this section. As shown in Figure 1, 10 design points are sampled and executed. The detailed data are shown in Table 1. Additive scaling function is used for both low-fidelity models and the polynomial degree is chosen as $p = 1$. The normalization parameters are $x_{max} = 0.9119$ and $x_{min} = 0.0625$. The model parameters estimated in Step 2 are $\theta_1 = 0.0028$ and $\theta_2 = 0.5434$.

	1	2	3	4	5	6	7	8	9	10
x	0.6140	0.4520	0.2287	0.8429	0.7857	0.5448	0.0625	0.9119	0.3315	0.1506
y_{l_1}	1.004	2.090	2.120	1.001	1.991	2.439	2.984	2.390	2.142	1.120
y_{l_2}	2.213	2.477	1.715	2.078	1.296	2.950	2.806	2.897	1.034	2.772
y_h	4.277	4.202	4.855	5.254	4.113	4.844	6.898	6.594	4.410	4.341

Table 1: The detail data of DOE

Given a new point $x = 0.25$ to be evaluated, the outputs of low-fidelity models at this unobserved point are $y_{l_1}(0.25) = 2.657$ and $y_{l_2}(0.25) = 1.428$. These two outputs are corrected at each design points and the corrected data (i.e., $\tilde{y}_i^{l_j}(x), \forall i = 1, \cdots, 10, j = 1, 2$) are represented in Figure 2 (the blue square marks and the yellow diamond marks in the left figure). Two polynomials are locally fitted around this new point (the blue dashed line and the yellow dotted line). The estimates at this new point provided by these two corrected low-fidelity surrogate models are the intersection points of the corresponding fitted polynomial and the new point line, which are $\hat{y}_{l_1}(0.25) = 5.301$ and $\hat{y}_{l_2}(0.25) = 4.562$.

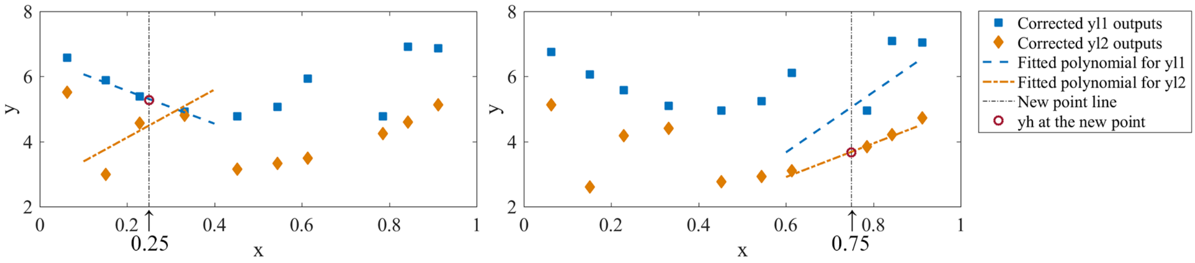


Figure 2: The fitted polynomials at the unobserved point $x = 0.25$ and $x = 0.75$.

The corrected outputs provided by y_{l_1} can much better be locally fitted compared to those provided by y_{l_2} according to the left figure in Figure 2. Consistent with this, $W\hat{S}E_{l_1}(0.25) = 0.0008$ is smaller than $W\hat{S}E_{l_2}(0.25) = 0.0913$. Thus, higher weight will be assigned to y_{l_1} (which is equal to 1 according to equation (12)). The final estimate is very close to the real value 5.280 (the red circle mark).

The right figure in Figure 2 shows the locally fitted polynomials for new point $x = 0.75$. In this area, low-fidelity model y_{l_2} performs better than model y_{l_1}, no matter according to the figure or according to the estimated weighted square error: $W\hat{S}E_{l_1}(0.75) = 0.3289$ and $W\hat{S}E_{l_2}(0.75) = 0.0007$.

1,000 checkpoints are equally sampled to be predicted. With the polynomial degree fixed ($p = 1$), the predictors considering different low-fidelity models are built as shown in Figure 3(a)(b)(c)(e). Kernel Regression (KR) considers only the high-fidelity data. Without the help of low-fidelity models, it only captures the quadratic term in the high-fidelity model (Figure 3(a)). EKR predictors considering only $\hat{y}_{l_1}$ (Figure 3(b)) or $\hat{y}_{l_2}$ (Figure 3(c)) can follow the general trend in one region but have poor behavior in the other. Compared with these two predictors, the EKR predictor containing both low-fidelity models (Figure 3(e)) can choose the right low-fidelity model in different areas and generate a good match of the real function among the whole domain. Larger errors occur at the boundary of the domain and at the middle where a suddenly change of low-fidelity models' importance levels happens.

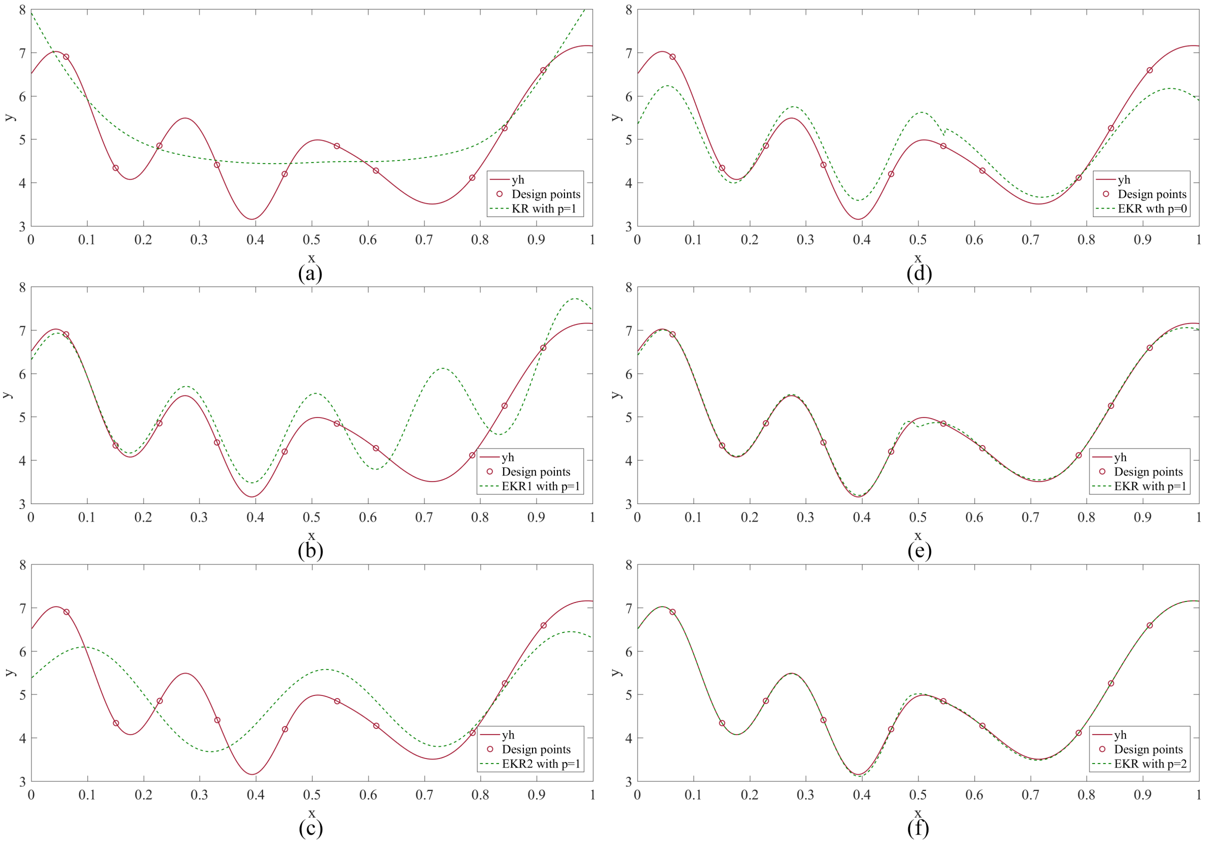


Figure 3: Estimates provided by different predictors compared with the real value. (a), Kernel Regression ($\theta = 0.0147$). (b) and (c), EKR with only y_{l_1} and only y_{l_2}, respectively. (d)(e)(f), EKR with both y_{l_1} and y_{l_2}, but different polynomial degree p.

When both low-fidelity models are considered, the EKR predictors with different polynomial degrees have different performance (Figure 3(d)(e)(f)). The predictor with $p = 0$ (Figure 3(d)) is not able to capture the trend of the quadratic bias term. The estimates are much lower than they should be at the boundary, while higher than the true values at the valley of the quadratic bias. The accuracy of the prediction is highly improved when the polynomial degree is increased to 1, while the improvement from $p = 1$ to $p = 2$ is not significant.

4.3 Numerical analysis

In this section, every experiment is executed with 50 replications. Each replication applies Algorithm 1 with the same model setting but the DOE points used to build the predictor are re-sampled. All the points are sampled by Latin Hypercube Sampling (Helton and Davis, 2003). For each built predictor, the same 1,000 check points are used for the integral prediction performance testing.

4.3.1 Main factors

Four factors are required to be decided before implementing the EKR method: type of scaling function, low-fidelity models, polynomial degree and DOE size. A complete factorial design is developed to analyze how these four factors affect the performance of EKR method. The type of scaling function has two levels: additive scaling function and multiplicative scaling function (same scaling function is used for both low-fidelity models). EKR model has three levels: "EKR1" (EKR model considering y_{l_1}), "EKR2" (EKR model considering y_{l_2}) and "EKR_both" (EKR model considering both low-fidelity models). Polynomial degree has three levels: $p = 0$, $p = 1$ and $p = 2$. DOE size has two levels: $n = 10$ points and $n = 20$ points. The Root Mean Square Error (RMSE) is used as the synthetic indicator of the prediction performance:

$$RMSE = \sqrt{\frac{1}{R}\sum_{i=1}^{R}(\hat{y}(x_i) - y_h(x_i))^2}, \tag{15}$$

where $R = 1,000$ checkpoints, $\hat{y}(x_i)$ and $y_h(x_i)$ are the estimate provided by the predictor and the high-fidelity response at checkpoints x_i, respectively.

As shown in Figure 4, the type of scaling function has the greatest effect on the estimation. The predictors with additive scaling function have significant improvement compared to the predictors with multiplicative scaling function in this case (the P-value is equal to 0.000 in

Mann-Whitney test). Thus, additive scaling function is used in the following analysis. However, scaling function is a case-to-case factor, it is not possible to generalize which one is better than others. The low-fidelity models contained in the EKR model have also significant effect on the estimation, as well as the DOE size. The influence of polynomial degree seems to be not significant compared to the other three factors.

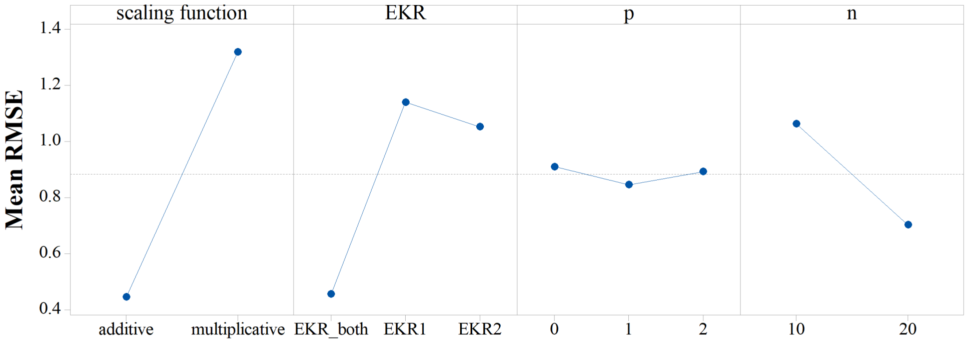


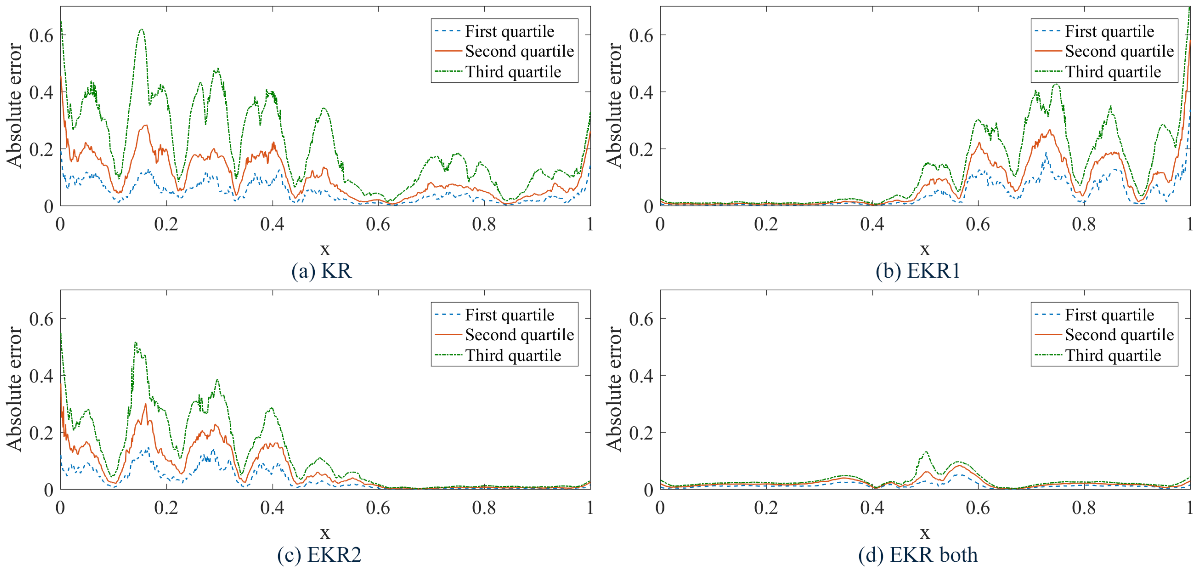
Figure 4: Main effect plot of the complete factorial design.

4.3.2 Detail Performance Analysis

With polynomial degree and DOE size fixed ($p = 1, n = 20$), the effect of the low-fidelity models considered in the EKR predictors is presented in Figure 5. At each checkpoint, we have 50 estimates from the 50 replications in each experiment. Figure 5 shows the first, second and third quartiles of these 50 estimates' absolute errors at each checkpoint.

As shown in Figure 5(a), the KR predictor almost misses all the detailed information of the system performance except the quadratic bias item (the smaller errors occur when the fitted quadratic function crosses over the true function). The EKR predictors considering single low-fidelity model have good behavior in one area while behave roughly in the others (Figure(b) and Figure(c)). The EKR predictor considering both low-fidelity models is accurate in estimating the fitted function except in the middle of domain where the importance levels of low-fidelity models change rapidly. In this case, more design points are required in the middle to perform well.

With both low-fidelity models considered, the effect of polynomial degree and DOE size is presented in Figure 6. For all p values, the accuracy and the precision of the predictor are improved (both the error and the variance decrease) when DOE size increases. The improvement from $p = 0$ to $p = 1$ is significant while the performance of predictors with $p = 1$ and $p = 2$ are similar.



(a) KR

(b) EKR1

(c) EKR2

(d) EKR both

Figure 5: The quartiles of the absolute errors of KR and EKR predictors in 50 replications. Polynomial degree $p = 1$ and DOE size $n = 20$. "EKR1" and "EKR2": EKR predictors considering only y_{l1} and only y_{l2}, respectively. "EKR_both": EKR predictor considering both low-fidelity models.

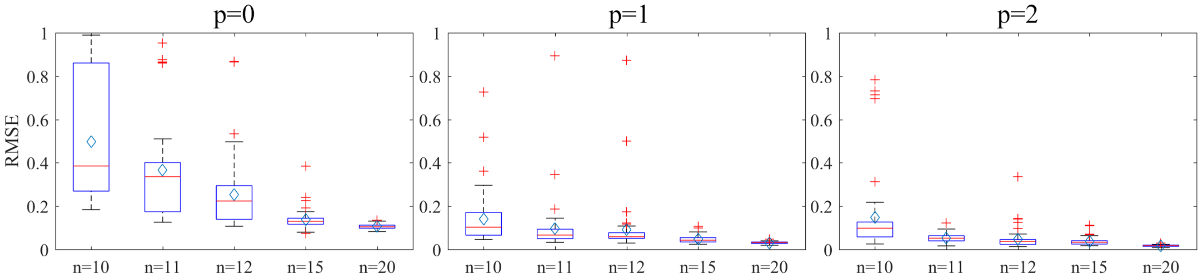


Figure 6: Boxplots of RMSE for EKR predictors with different DOE size n and different polynomial degree p. Both low-fidelity models are considered in the predictors. Each boxplot contains 50 replications.

The left figure in Figure 7 shows the average of the automatically calculated weights of low-fidelity model y_{l_1} at each checkpoint. The weights of y_{l_1} converge when DOE size increases properly. The automatically calculated weights of low-fidelity models are the importance of the low-fidelity models for the prediction. They could be different for different polynomial degrees. For predictors with $p = 0$, the weights of y_{l_1} climbs when $x > 0.9$ because of the effect of the quadratic bias term. For predictors with polynomial degree $p = 1$ and $p = 2$, the weights of y_{l_1} calculated by EKR method show a reversed "S"-shape as expected.

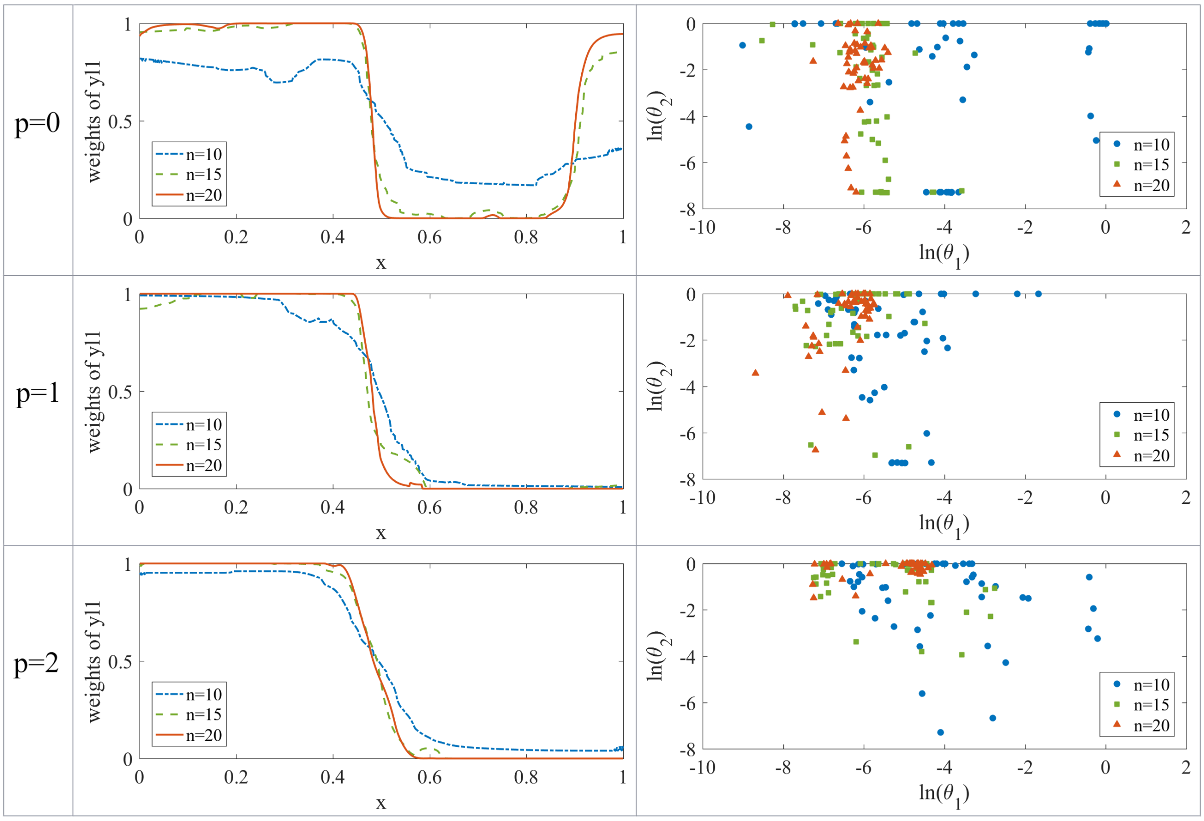


Figure 7: The averages of calculated weights of low-fidelity model y_{l_1} at each checkpoint (left) and the selected θ_1, θ_2 value (right) with different DOE size n and different polynomial degree p. 50 replications are executed.

The right figure in Figure 7 shows in logarithmic scale the θ_1 and θ_2 values selected in the 50 replications as described in section 3.2.1 (θ_1 is a scalar since this is a single-dimensional case). The selected θ_1 and θ_2 values are spread in a large area when DOE size is small (10 design points) while have smaller variances when DOE size is large (20 design points). θ_1 value slightly decreases as DOE size increases. A smaller local area is considered in the predictor when DOE size increases because more data are available. For predictors with $p = 1$ and $p = 2$, θ_2 value converges to a small value as DOE size increases. However, for predictors with $p = 0$, the speed of converging is slow.

4.3.3 Prediction Interval

An example of prediction interval is shown in Figure 8. The DOE size is 20 design points, $p = 1$ and the confidence level is $1 - \alpha = 0.95$. These prediction intervals catch 96.8% of the 1,000 check points' true value. From the figure, we can find the length of the prediction interval is short except in the middle of domain where the proposed EKR predictor might have larger prediction bias. The prediction interval length at the peak or the valley of the fitted function is relative larger than that in other areas.

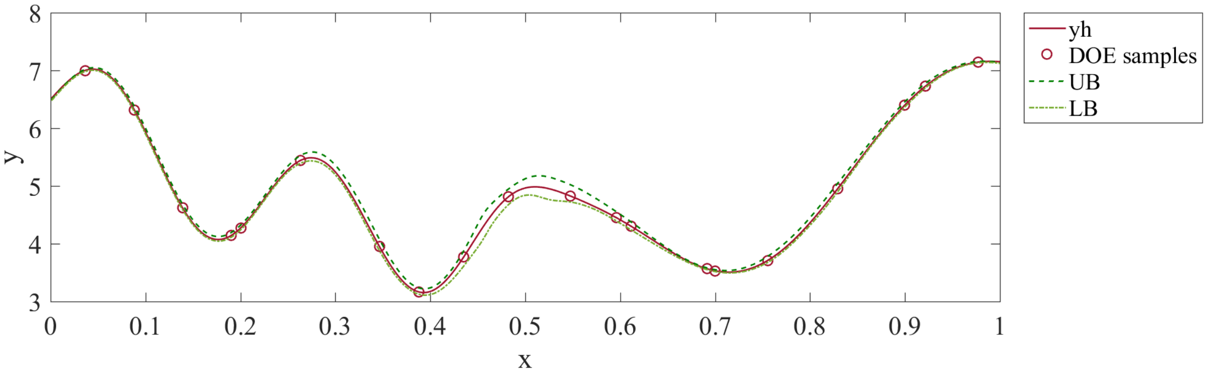


Figure 8: The upper bound (UB) and the lower bound (LB) of the proposed prediction interval with confidence level $1 - \alpha = 0.95$. 20 points are contained in the DOE.

Figure 9 shows the ratio of the 1,000 check points caught by the provided prediction interval with $p = 1$ and different confidence levels over 50 experiments. We can see that the mean ratio converges to 0.95 and the variance decreases as the size of the design points increases when $1 - \alpha = 0.95$. Even though the ratio is a little higher than the selected confidence level when $1 - \alpha < 0.95$, the ratio decreases as the confidence level reduces.

4.3.4 Robustness to wrong low-fidelity models

It is not always possible to know in advance if the selected low-fidelity models are helpful for the prediction. Even worse, they could provide wrong information. For example, let us consider as low-fidelity model the following function:

$$y_{l_3}(x) = \sin(30x + 1) + 3.$$

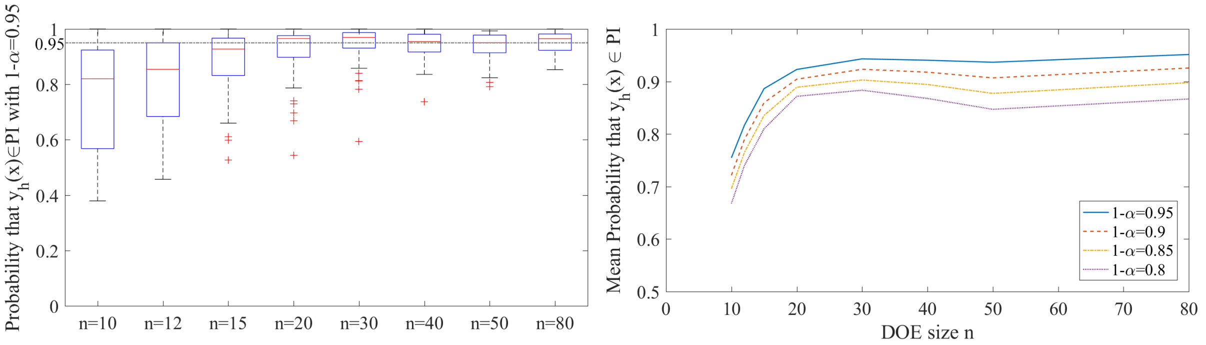


Figure 9: The ratio of the 1,000 check points caught by the prediction interval with confidence level $1 - \alpha = 0.95$ (left) and the mean ratio of 50 replications with different confidence levels (right). 50 replications are executed.

As shown in Figure 10, it seems that y_{l_3} does not contain any structural property of the true function, a wrong general trend is also provided. However, this issue could not be known in advance. In this subsection, we investigate how the proposed method behaves when a wrong model is adopted as low-fidelity model.

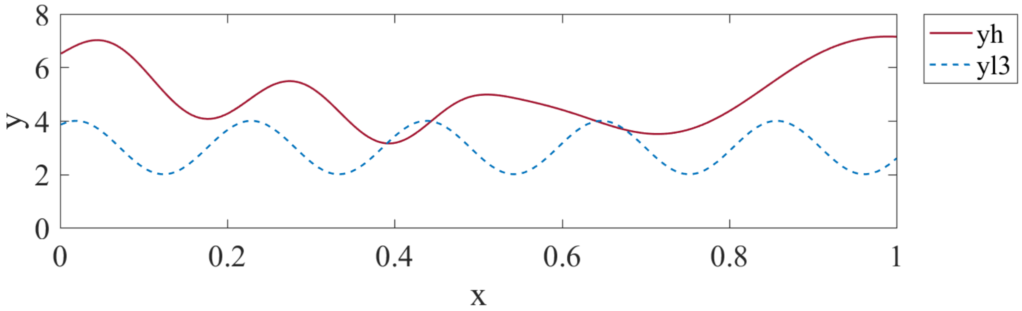


Figure 10: The true function y_h and a wrong low-fidelity model y_{l_3}.

In the EKR method, a special low-fidelity model can be considered together with the selected low-fidelity models (i.e., y_{l_3} in this test case) :

$$y_{l_0}(\boldsymbol{x}) \equiv c,$$

where c is a constant and $c \neq 0$ for multiplicative scaling function. The corrected outputs of y_{l_0} are the high-fidelity responses at design points: $\tilde{y}_i^{l_0}(\boldsymbol{x}) = y_h(\boldsymbol{x}_i^0), \forall i \in \mathcal{N}$. Thus, the corrected low-fidelity surrogate model $\hat{y}_{l_0}(\boldsymbol{x})$ is the pure Kernel Regression predictor. The proposed method will compare these corrected outputs with other corrected outputs provided by the selected low-fidelity models and combine them according to their local prediction errors. If the highest weight is assigned to y_{l_0} in one area, it means that considering only high-fidelity data has higher accuracy for the prediction than combining high-fidelity data with any of the selected low-fidelity models in this area.

Figure 11 presents the RMSE for KR predictor and EKR predictors with or without y_{l_0} as DOE size increases. Additive scaling function is used and $p = 1$. A first comment is that, independently from the low-fidelity models used in the prediction, RMSE decreases as the sample size increases. This means that the prediction improves with increase of data despite the quality of the low-fidelity models adopted. The prediction performance of EKR is worse than KR because of the wrong low-fidelity model considered. The involvement of y_{l_0} can reduce the influence of the wrong low-fidelity model and help the EKR predictor to converge to KR predictor faster than the EKR predictor considering only the wrong low-fidelity model. In Figure 11 it is possible to notice that, with a sample size of 25, the EKR predictor considering both y_{l_3} and y_{l_0} behaves as the KR predictor using only high-fidelity data, this is because the model starts assigning zero weight to the wrong low-fidelity model y_{l_3}.

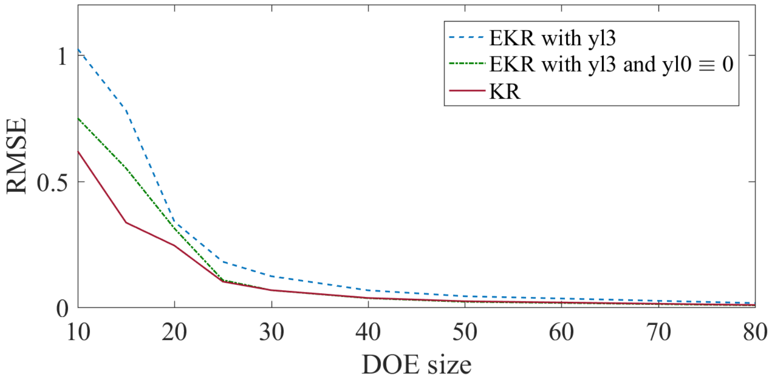


Figure 11: The mean RMSE of 50 replications for EKR predictor considering only y_{l_3}, EKR predictor considering both y_{l_3} and $y_{l_0} \equiv 0$ and KR predictor. Polynomial degree $p = 1$ and additive scaling function is used.

The involvement of the constant low-fidelity model can improve the robustness to wrong low-fidelity models. Nevertheless, if some helpful low-fidelity models are selected, its involvement might not fully exploit information from helpful low fidelity models, which could reduce the efficiency of the algorithm. Thus, the constant low-fidelity model is suggested to be used in preliminary exploratory analysis to check the performance of the low-fidelity models. Then, a selection of which low fidelity models might be helpful for the prediction can be done.

5 Application

The proposed method is applied to the closed-loop flexible assembly system (CLFAS) studied in Suri and Leung (1987). Two low-fidelity models and one high-fidelity model are considered in the system performance evaluation.

5.1 System description

In CLFAS, several workstations are connected together in a loop. The buffer capacities between adjacent workstations are limited and blocking after service is applied. The workpieces are loaded on pallets and the number of pallets in the entire system is fixed. The workpieces can only be loaded and unloaded at the first workstation, which makes the first workstation the only exit of the system. It is assumed that a large number of workpieces are waiting outside the system, once a finished workpiece leaves the system, a new workpiece will enter the system to be processed. Thus, the number of workpieces assembled in the system is fixed.

As shown in Figure 12, the system has six workstations and only one buffer slot between adjacent workstations. The number of pallets in the system is also six. The transfer time between two workstations is assumed negligible. For the convenience of analysis, we assume that this system only contains two kinds of machines M_1, M_2. The processing time of each machine is independent and follows Gamma distribution:

$$T_r \sim \Gamma\left(\frac{1}{scv_r}, \frac{1}{x_r scv_r}\right), \forall r = 1, 2,$$

where x_r is the mean processing time of machine M_r and scv_r is the squared coefficient of variation (SCV) of the processing time of machine M_r. The machines in the first three workstations are M_1 type and the machines in the last three workstations are M_2 type. Given any combination of (x_1, x_2, scv_1, scv_2) within the predefined domain, we want to estimate the expected throughput of the system (TH).

5.1.1 Experimental Design

The system configuration contains the mean processing times and the SCVs of M_1 and M_2: $\boldsymbol{x} = (x_1, x_2, scv_1, scv_2)$. The domain of interest is $x_r \in [0.05, 0.15], scv_r \in (0, 0.15], \forall r = 1, 2$. Two analytical methods are applied in this case as low-fidelity models. A Continuous Time Markov Chain (CTMC), which assumes the processing time at each workstation T_r follows an

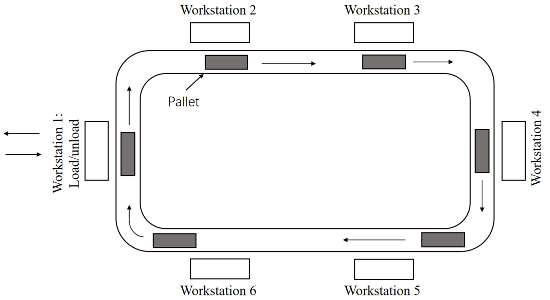


Figure 12: Closed-Loop Flexible Assembly System with six workstations and six pallets.

exponential distribution (i.e., $scv_r = 1, \forall r = 1, 2$). The second method is bottleneck based and calculates the throughput as follows:

$$TH = 1/\max\{x_1, x_2\},$$

assuming that processing time at each workstation T_r is deterministic ($scv_r = 0, \forall r = 1, 2$). A simulation model is built as the high-fidelity model with $1,000,000$ workpieces of simulation length and $500,000$ workpieces of warm-up period. The simulation length is long enough so that the half length of the confidence interval is smaller than 0.1%.

Four design sizes are developed (10, 20, 30 and 50 points in the DOE) and each experiment is repeated 50 times (each replication has the same DOE size but different sample points). All the points are sampled by Latin Hypercube Sampling (Helton and Davis, 2003). The proposed EKR method as well as KBNN method and Kriging predictor are applied in this case.

In EKR method, the polynomial degree $p = 1$ is used and multiplicative scaling function is used for both low-fidelity models. The choice of the scaling function was decided by explorative data analysis. For the sake of simplicity, we assume the same type of system configuration components have the same θ_1 value (i.e., $\theta_{11} = \theta_{12}, \theta_{13} = \theta_{14}$).

In KBNN method (Wang and Zhang, 1997), low-fidelity models are put in the knowledge layer as active functions with two neurons. Linear boundary functions are used in boundary layer and logistic function is used in region layer. The configuration 2 neurons in boundary layer and 9 neurons in region layer is used. The number of neurons is decided through a complete factorial design with 2 factors ("number of boundary neurons" and "number of region neurons"). The factor "number of boundary neurons" has 5 levels:{1,2,3,4,5} and the factor "number of region neurons" has 10 levels:{2,3,4,5,6,7,8,9,10,11}. The complete factorial design is executed when DOE size is equal to 50. 50 replications are executed. Trust region algorithm with gradient is used to train this neural network with the design points as training set.

In Kriging predictor, the DACE code available in (Lophaven et al., 2002) is used, modified in the basis functions which are chosen as low-fidelity models. The model of the Kriging method is as follows:

$$y_h(\boldsymbol{x}) = \beta_1 y_{l_1}(\boldsymbol{x}) + \beta_2 y_{l_2}(\boldsymbol{x}) + z(\boldsymbol{x}),$$

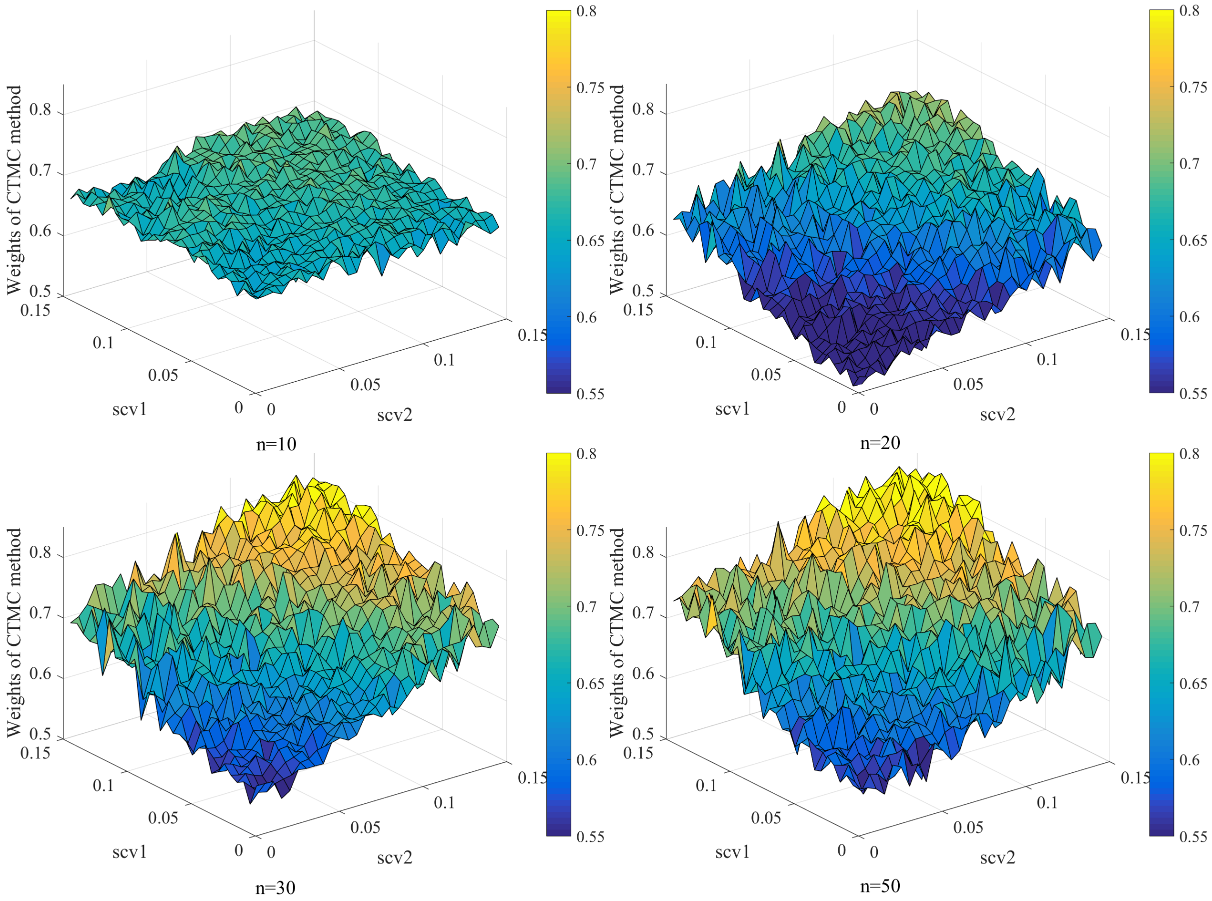
where $y_{l_1}(\boldsymbol{x})$ is CTMC method, $y_{l_2}(\boldsymbol{x})$ is the bottleneck-based method and $z(\boldsymbol{x})$ is a stationary Gaussian Process. The correlation function is chosen as Gaussian function. The initial values, lower bounds and upper bounds of model parameters are $[0.01]^4$, $[0.00001]^4$ and $[10]^4$, respectively. The Kriging predictor can automatically calculate the weights of low-fidelity models (i.e., β_1, β_2) using Maximum Likelihood method and they are kept the same among the whole domain.

R=10,000 checkpoints are generated and the RMSE in equation (15) is used to evaluate the predictors' (i.e., EKR, KBNN and Kriging) performance.

5.1.2 Numerical Results

Figure 13 shows the automatically calculated weights of CTMC method in EKR predictors. To have a better understanding of the data, the *scv*1 and *scv*2 are divided into 30 equal slots and the data shown in Figure 13 is the average weight in each region constructed by combining *scv*1 and *scv*2 slots.

The weights of CTMC method are expected to have large value when the variability of the system is large (i.e., both *scv*1 and *scv*2 are large) and have small value when both *scv*1 and *scv*2 are small. As shown in Figure 13, when DOE size is small (i.e., n=10), the EKR predictor does not perform as expected, because the collected information is too scarce. The importance of each low-fidelity model almost stays the same during the whole domain. When DOE size increases to 20 points, the weights of CTMC method start to present a trend of raise as the variability of the system increases. When DOE size increases to 30 and 50 points, a significant difference between the weights of CTMC method in different areas can be found. The EKR method has the ability to choose reliable low-fidelity models in different areas if enough information is contained in the model. The weights of low-fidelity models will become stable when enough information is collected (there is not much difference between these weights when n=30 and when n=50).



n=10

n=20

n=30

n=50

Figure 13: The automatically calculated weights of CTMC method in the predictors built with different DOE size n.

We can also notice that, although the importance levels of these two low-fidelity models are changing among the whole domain, the CTMC method has always higher importance than the bottleneck method (weights of CTMC are always larger than 0.5). This implies that the CTMC method always contains more helpful information than the bottleneck method according to EKR method's judgement.

Three kinds of EKR models are applied in this case: the EKR model built with one of the two low-fidelity models and the EKR model built using both low-fidelity models. Also, KR model considering no low-fidelity models is applied. As shown in Figure 14, KR predictor requires lots of design points to perform well compared to EKR predictors. The prediction error of "EKR1" (i.e., EKR model considering only CTMC method) is always smaller than "EKR2" (i.e., the EKR model considering only the bottleneck method). This is consistent with the conclusion we drawn from the calculated weights of low-fidelity models. The "EKR" predictor (i.e., the EKR model using both low-fidelity models) is always better than KR predictor and the other two EKR predictors as expected. The accuracy of these four predictors is improved as DOE size increases, as well as the variability.

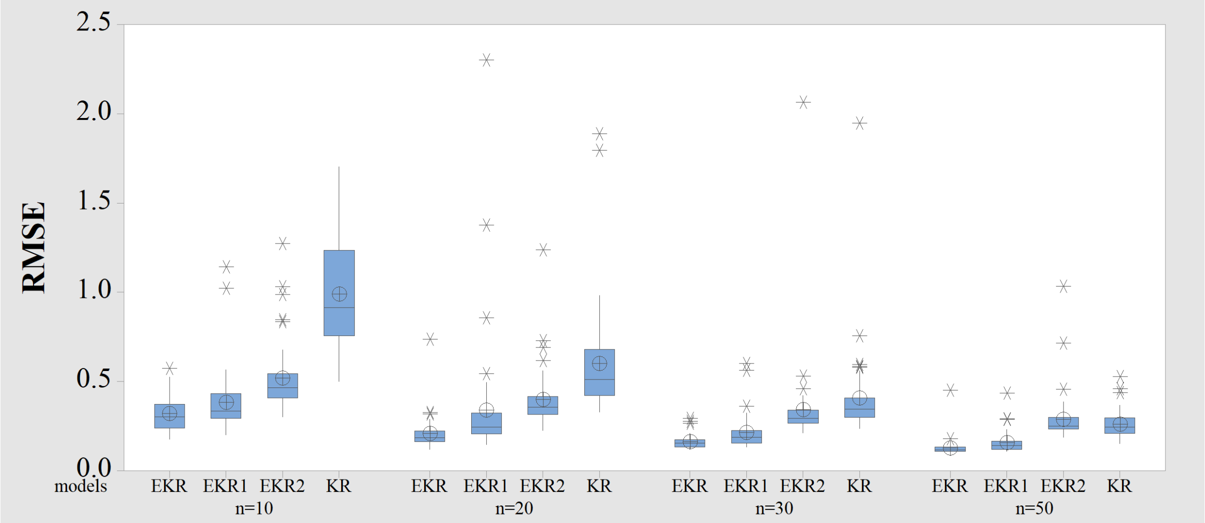


Figure 14: Boxplots of RMSE for EKR predictors and KR predictors with different DOE size n. "EKR1" and "EKR2" denote the EKR models are built with only CTMC method and only the bottleneck method, respectively. "EKR" denotes EKR model is built with both low-fidelity models. Each boxplot contains 50 replications.

The EKR method is numerically compared with KBNN method and Kriging predictor. To make the comparison fair enough, the CTMC method and the bottleneck method are both considered in these three methods. As shown in Figure 15, the prediction errors of these three predictors decrease as DOE size increases, as well as their variance. The KBNN predictor does not perform well in this case. The reason could be related to the high number of parameters to be fit during the training. In this case, 65 parameters are fitted. When the DOE size is scarce (10 points), EKR predictor assigns almost similar weights to both low-fidelity models, the gap between the performance of EKR and Kriging is small. As DOE size increases, a visible improvement of EKR predictor can be found for both the mean error and the variability. This is because the EKR method has enough information to select helpful low-fidelity model in different areas while the importance of low-fidelity models in Kriging is kept the same during the whole domain.

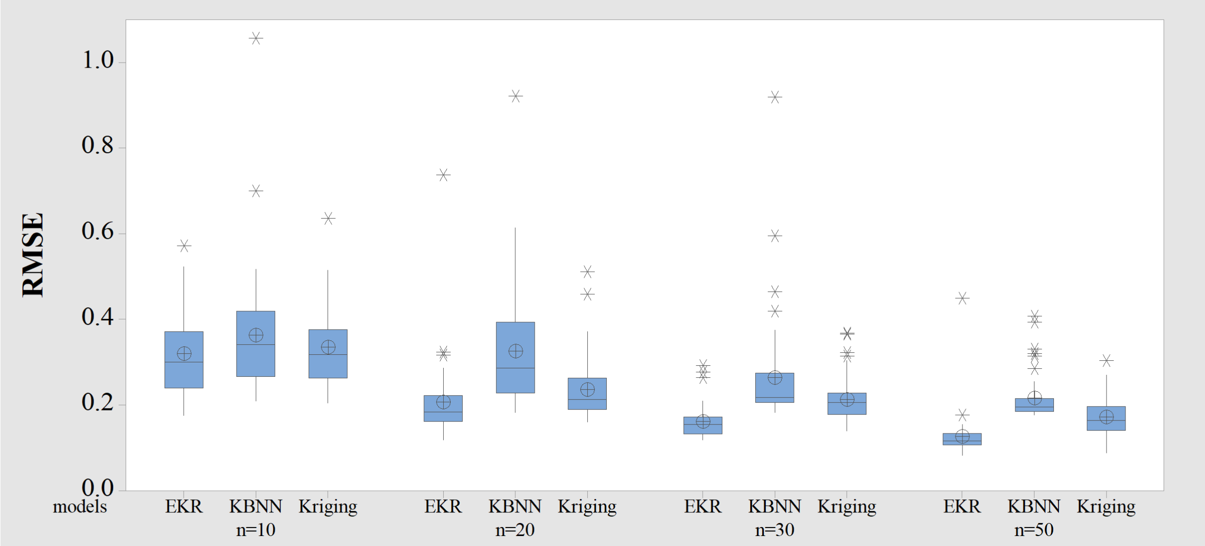


Figure 15: Boxplots of RMSE for EKR, KBNN and Kriging models with different DOE size n. Both low-fidelity models are considered in these predictors. Each boxplot contains 50 replications.

Figure 16 shows the ratio of the 10,000 checkpoints caught by the prediction intervals. The mean relative half-length of prediction interval with $1 - \alpha = 0.95$ is smaller than 2% for all designs and it decreases as DOE size increases. Similar as in the illustrative example, the mean ratio converges to 0.95 and the variance reduces as the size of the design points increases. The mean ratio decreases when the confidence level decreases.

6 Conclusion

A method combining simulation experiments with analytical models for system performance evaluation is proposed in this paper. This method combines high-fidelity model and low-fidelity

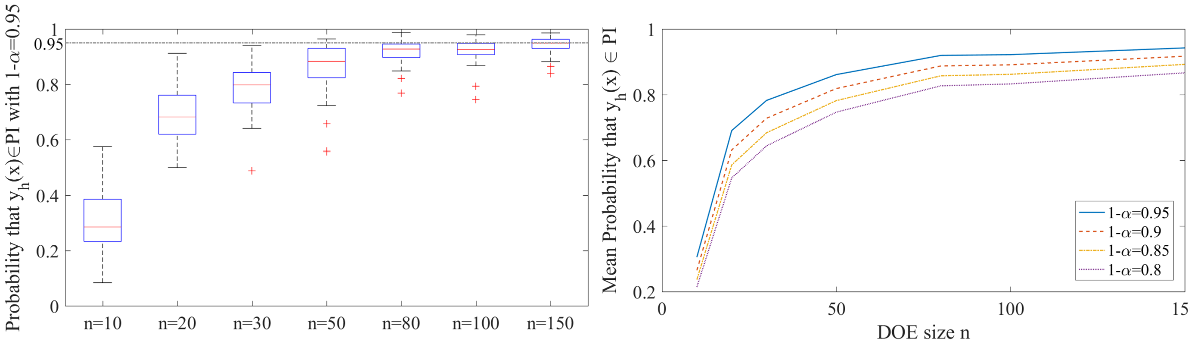


Figure 16: The ratio of the 10,000 check points caught by the prediction intervals with confidence level $1 - \alpha = 0.95$ (left) and the mean ratio of 50 replications with different confidence levels (right). 50 replications are executed.

models to have more accurate estimates rapidly. The method presents two main technical features: multiple non-hierarchical low-fidelity models can be considered, and the relative importance of low-fidelity models in predicting the performance is calculated only from data.

The proposed method is tested numerically in a closed-loop flexible assembly system and compared with alternative predictors. The results show that EKR method is capable of identifying the reliability of different low-fidelity models in different areas and combines them efficiently. This feature was significant for reaching good prediction performance in the tested application.

The EKR method is a regression technique that can be applied in both stochastic and deterministic problems. The method is not able to interpolate points when deterministic functions are fitted. This is a limitation of the proposed approach.

The EKR method can be further improved in different ways. Low-fidelity models are used only at design points where high-fidelity experiments are also available. Using low-fidelity models in other points but DOE points could be useful to improve prediction. For instance, Co-Kriging technique allows this assuming a certain covariance function. Another direction of improvement is to consider the possibility that design points might be not balanced. In this situation, variable $\boldsymbol{\Theta}_1(\boldsymbol{x})$ values might be helpful to increase the accuracy of the prediction. Currently, the $\boldsymbol{\Theta}_1$ values are the same during the whole domain. In addition, variable $\boldsymbol{\Theta}_1(\boldsymbol{x})$ might be helpful for interpolating design points in deterministic case. Small $\boldsymbol{\Theta}_1(\boldsymbol{x}_i^0)$ narrows the region considered as local area when evaluating the i-th design point, which could force the predictor to almost interpolate $(\boldsymbol{x}_i^0, y_h(\boldsymbol{x}_i^0))$.

Despite Kernel Regression is used to build the predictor, alternative approaches can be developed using the same concept of adjusting the weights of low-fidelity models in different areas by extending other regression or interpolating techniques like Kriging.

Further efforts will be put on developing a meta-model based simulation optimization algorithm with an efficient dynamic sampling methodology for the proposed EKR model.

Appendix A Prediction Interval for Linear Fitting

According to Wand and Jones (1995), if the following assumptions hold:

1. The density function of the design points $f(\boldsymbol{x})$, the variance of the observations $v(\boldsymbol{x}) = \mathrm{Var}(y(\boldsymbol{x}))$ and all entries of the true function's Hessian Matrix $\boldsymbol{H}_{Ey}(\boldsymbol{x})$ are continuous;
2. $\boldsymbol{\Theta} = \boldsymbol{\Theta}_n$ is a sequence of bandwidth matrices such that $n^{-1}|\boldsymbol{\Theta}|^{-1/2}$ and all entries of $\boldsymbol{\Theta}$ approach zero as $n \to \infty$, also, the ratio of the largest and smallest eigenvalues of $\boldsymbol{\Theta}$ is bounded for all n;
3. The size of design points n is sufficiently enough and the unknown point is not near the boundary of the design;

the bias and the variance of local linear regression predictor (i.e., polynomial degree $p = 1$) given unknown point $\boldsymbol{x}$ are:

$$\mathrm{E}(\hat{y}(\boldsymbol{x})) - \mathrm{E}(y(\boldsymbol{x})) = \frac{1}{2}\mu_2(K)\mathrm{tr}(\boldsymbol{\Theta}\boldsymbol{H}_{Ey}(\boldsymbol{x})) + o_P(\mathrm{tr}(\boldsymbol{\Theta})), \tag{16}$$

$$\mathrm{Var}(\hat{y}(\boldsymbol{x})) = \frac{R(K)v(\boldsymbol{x})}{n|\boldsymbol{\Theta}|^{1/2}f(\boldsymbol{x})} + o\left(\frac{1}{n|\boldsymbol{\Theta}|^{1/2}}\right), \tag{17}$$

where K is a symmetric kernel function satisfying $\int K(\boldsymbol{u})d\boldsymbol{u} = 1$, $R(K) = \int K(\boldsymbol{u}; \boldsymbol{\Theta} = \boldsymbol{I})^2 d\boldsymbol{u}$ and $\mu_2(K) = \int u_k^2 K(\boldsymbol{u}; \boldsymbol{\Theta} = \boldsymbol{I})d\boldsymbol{u}$ is independent of $k = 1, \cdots, d$.

Back to the proposed EKR model, assuming that the bandwidth matrix $\boldsymbol{\Theta}_1$ and θ_2 are selected, we would like to provide a prediction interval for the performance of a given unknown point $\boldsymbol{x}$, rather than just provide the mean prediction. Let

$$\tilde{y}_{l_j}(\boldsymbol{u}; \boldsymbol{x}) = y_{l_j}(\boldsymbol{x}) + (y_h(\boldsymbol{u}) - y_{l_j}(\boldsymbol{u})) \quad \text{or} \quad \tilde{y}_{l_j}(\boldsymbol{u}; \boldsymbol{x}) = \frac{y_h(\boldsymbol{u})}{y_{l_j}(\boldsymbol{u})} \cdot y_{l_j}(\boldsymbol{x}),$$

and

$$\tilde{y}(\boldsymbol{u}; \boldsymbol{x}) = \sum_{j \in \mathcal{J}} w_{l_j}(\boldsymbol{x}) \tilde{y}_{l_j}(\boldsymbol{u}; \boldsymbol{x}).$$

Using the EKR predictor defined in equation (11) and equation (7) we obtain:

$$\begin{aligned}\hat{y}_{EKR}(\boldsymbol{x}) &= \sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})\hat{y}_{l_j}(\boldsymbol{x})\\ &= \sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})\boldsymbol{e}_1^T(\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{X}_{\boldsymbol{x}})^{-1}\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W}_{\boldsymbol{x}}\tilde{\boldsymbol{Y}}_{l_j}\\ &= \boldsymbol{e}_1^T(\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{X}_{\boldsymbol{x}})^{-1}\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W}_{\boldsymbol{x}}\tilde{\boldsymbol{Y}},\end{aligned}$$

where

$$\tilde{\boldsymbol{Y}} = [\tilde{y}(\boldsymbol{x}_1^0;\boldsymbol{x}),\cdots,\tilde{y}(\boldsymbol{x}_n^0;\boldsymbol{x})]^T,$$

is a predictor fitting the weighted average of the corrected low-fidelity outputs $\tilde{y}(\boldsymbol{x}_i^0;\boldsymbol{x}), \forall i \in \mathcal{N}$ by Kernel Regression. We know that $\mathrm{E}(\tilde{y}(\boldsymbol{x};\boldsymbol{x})) = \mathrm{E}(y_h(\boldsymbol{x}))$ and $\mathrm{Var}(\tilde{y}(\boldsymbol{x};\boldsymbol{x})) = \mathrm{Var}(y_h(\boldsymbol{x}))$ due to the deterministic assumption on low-fidelity models in section 2. Assuming the above three assumptions hold, expressions (16) and (17) become:

$$\begin{aligned}\mathrm{E}(\hat{y}_{EKR}(\boldsymbol{x})) - \mathrm{E}(y_h(\boldsymbol{x})) &= \mathrm{E}(\hat{y}_{EKR}(\boldsymbol{x})) - \mathrm{E}(\tilde{y}(\boldsymbol{x};\boldsymbol{x}))\\ &= \frac{1}{2}\mu_2(K_1')\mathrm{tr}(\boldsymbol{\Theta}_1\boldsymbol{H}_{E\tilde{y}}(\boldsymbol{x})) + o_P(\mathrm{tr}(\boldsymbol{\Theta}_1)),\end{aligned} \tag{18}$$

$$\mathrm{Var}(\hat{y}_{EKR}(\boldsymbol{x})) = \frac{R(K_1')v(\boldsymbol{x})}{n|\boldsymbol{\Theta}_1|^{1/2}f(\boldsymbol{x})} + o\left(\frac{1}{n|\boldsymbol{\Theta}_1|^{1/2}}\right), \tag{19}$$

where $\boldsymbol{H}_{E\tilde{y}}(\cdot)$ is the Hessian matrix of the function $\mathrm{E}(\tilde{y}(\cdot;\boldsymbol{x}))$, $f(\cdot)$ is the density function of sampled points and $v(\boldsymbol{x}) = \mathrm{Var}(\tilde{y}(\boldsymbol{x};\boldsymbol{x})) = \mathrm{Var}(y_h(\boldsymbol{x}))$. $K_1'(\cdot)$ is $K_{1,\boldsymbol{\Theta}}$ kernel function modified by multiplying a constant:

$$K_1'(\cdot) = (2\pi)^{-d/2}|\boldsymbol{\Theta}_1|^{-1/2}K_{1,\boldsymbol{\Theta}_1}(\cdot),$$

so that $\int K_1'(\boldsymbol{u})d\boldsymbol{u} = 1$.

Since the expected bias is related to the Hessian Matrix of an unknown function which cannot be estimated well, it is more convenient to model the bias of the predictor as the variance of the true function that is easier to estimate. We assume the function $\tilde{y}(\cdot;\boldsymbol{x})$ is linear in the local area containing $\boldsymbol{x}$ and $v(\boldsymbol{x}) = \mathrm{E}\left((\tilde{y}(\boldsymbol{x};\boldsymbol{x}) - \mathrm{E}(\hat{y}_{EKR}(\boldsymbol{x})))^2\right) = \mathrm{Var}(y_h(\boldsymbol{x})) + bias^2$, i.e., the variance is calculated considering both the real variance of the true function $\tilde{y}(\cdot;\boldsymbol{x})$ and the bias of the predictor. Then, the left term in equation (18) is equal to zero and the bias is moved into equation (19). As $|\boldsymbol{\Theta}| \to 0, n|\boldsymbol{\Theta}|^{1/2} \to \infty$, applying the Central limit theorem, we have:

$$(\hat{y}_{EKR}(\boldsymbol{x}) - \mathrm{E}(y_h(\boldsymbol{x}))) \sim N\left(0, \frac{R(K_1')v(\boldsymbol{x})}{n|\boldsymbol{\Theta}_1|^{1/2}f(\boldsymbol{x})}\right), \tag{20}$$

and

$$(\hat{y}_{EKR}(\boldsymbol{x}) - y_h(\boldsymbol{x})) \sim N\left(0, v(\boldsymbol{x})\left(1 + \frac{R(K_1')}{n|\boldsymbol{\Theta}_1|^{1/2}f(\boldsymbol{x})}\right)\right). \tag{21}$$

The variance $v(\boldsymbol{x})$ can be estimated using the weighted square error of the whole predictor similar as equation (9):

$$\hat{v}(\boldsymbol{x}) = W\hat{S}E(\boldsymbol{x}) = (\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{x}}))^{-1}\tilde{\boldsymbol{Y}}^T(\boldsymbol{W}_{\boldsymbol{x}} - \boldsymbol{W}_{\boldsymbol{x}}^T\boldsymbol{X}_{\boldsymbol{x}}(\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{X}_{\boldsymbol{x}})^{-1}\boldsymbol{X}_{\boldsymbol{x}}^T\boldsymbol{W}_{\boldsymbol{x}})\tilde{\boldsymbol{Y}},$$

and the density function can be estimated using Kernel Density Estimator (Wand and Jones, 1995):

$$\hat{f}(\boldsymbol{x}) = \frac{1}{n}\sum_{i\in\mathcal{N}} K_1'(\boldsymbol{x}_i^0 - \boldsymbol{x}) = \frac{\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{x}})}{n(2\pi)^{d/2}|\boldsymbol{\Theta}_1|^{1/2}}.$$

We have $R(K_1') = (2\sqrt{\pi})^{-d}$ for Gaussian Kernel function, thus, equation (21) becomes:

$$(\hat{y}_{EKR}(\boldsymbol{x}) - y_h(\boldsymbol{x})) \sim N(0, \hat{s}(\boldsymbol{x})^2)$$

where

$$\hat{s}(\boldsymbol{x})^2 = W\hat{S}E(\boldsymbol{x})\left(1 + \frac{1}{2^{d/2}\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{x}})}\right),$$

and the prediction interval with confidence level $1 - \alpha$ is:

$$y_h(\boldsymbol{x}) \in [\hat{y}_{EKR}(\boldsymbol{x}) - Z_{\alpha/2}\hat{s}(\boldsymbol{x}), \hat{y}_{EKR}(\boldsymbol{x}) + Z_{\alpha/2}\hat{s}(\boldsymbol{x})]$$

where $Z_{\alpha/2}$ is the quantile value of standard normal distribution.

Appendix B Convergence for Linear Fitting

If the three assumptions hold, the absolute bias of the EKR predictor with $p = 1$ in equation (18) is:

$$\begin{aligned}
|\mathrm{E}(\hat{y}_{EKR}(\boldsymbol{x})) - \mathrm{E}(y_h(\boldsymbol{x}))| &= \frac{1}{2}\mu_2(K_1')|\mathrm{tr}(\boldsymbol{\Theta}_1 \boldsymbol{H}_{E\tilde{y}}(\boldsymbol{x}))| + o_P(\mathrm{tr}(\boldsymbol{\Theta}_1)) \\
&= \frac{1}{2}\mu_2(K_1')\left|\mathrm{tr}\left(\boldsymbol{\Theta}_1 \sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})\boldsymbol{H}_{E\tilde{y}_{l_j}}(\boldsymbol{x})\right)\right| + o_P(\mathrm{tr}(\boldsymbol{\Theta}_1)) \\
&= \frac{1}{2}\mu_2(K_1')\left|\sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})\mathrm{tr}(\boldsymbol{\Theta}_1 \boldsymbol{H}_{E\tilde{y}_{l_j}}(\boldsymbol{x}))\right| + o_P(\mathrm{tr}(\boldsymbol{\Theta}_1)) \\
&\leq \frac{1}{2}\mu_2(K_1')\sum_{j\in\mathcal{J}} w_{l_j}(\boldsymbol{x})|\mathrm{tr}(\boldsymbol{\Theta}_1 \boldsymbol{H}_{E\tilde{y}_{l_j}}(\boldsymbol{x}))| + o_P(\mathrm{tr}(\boldsymbol{\Theta}_1)) \\
&\leq \frac{1}{2}\mu_2(K_1')\max_j\{|\mathrm{tr}(\boldsymbol{\Theta}_1 \boldsymbol{H}_{E\tilde{y}_{l_j}}(\boldsymbol{x}))|\} + o_P(\mathrm{tr}(\boldsymbol{\Theta}_1)).
\end{aligned}$$

$\max_j\{|\mathrm{tr}(\boldsymbol{\Theta}_1 \boldsymbol{H}_{E\tilde{y}_{l_j}}(\boldsymbol{x}))|\} \to 0$ as $\boldsymbol{\Theta}_1 \to \mathbf{0}$, since $|\mathrm{tr}(\boldsymbol{\Theta}_1 \boldsymbol{H}_{E\tilde{y}_{l_j}}(\boldsymbol{x}))| \to 0, \forall j \in \mathcal{J}$ as $\boldsymbol{\Theta}_1 \to \mathbf{0}$. Therefore, the bias of the proposed predictor can converge to 0 if the size of the design points converges to infinite:

$$n|\boldsymbol{\Theta}_1|^{1/2} \to \infty, \boldsymbol{\Theta}_1 \to \mathbf{0} \text{ as } n \to \infty, \text{ and } \lim_{\boldsymbol{\Theta}_1 \to \mathbf{0}} |\mathrm{E}(\hat{y}_{EKR}(\boldsymbol{x})) - \mathrm{E}(y_h(\boldsymbol{x}))| = 0.$$

Then, as the size of the design points increases, the estimated variance will converge to the variance of the high-fidelity model for stochastic case and converge to 0 for deterministic case.

References

Askin, R. G. and Standridge, C. R. (1993). *Modeling and Analysis of Manufacturing Systems.* Wiley New York.

Bakr, M. H., Bandler, J. W., Madsen, K., Rayas-Sánchez, J. E., and Sondergaard, J. (2000). Space-mapping optimization of microwave circuits exploiting surrogate models. *IEEE Transactions on Microwave Theory and Techniques*, 48(12):2297–2306.

Bandler, J. W., Biernacki, R. M., Chen, S. H., Grobelny, P. A., and Hemmers, R. H. (1994). Space mapping technique for electromagnetic optimization. *IEEE Transactions on Microwave Theory and Techniques*, 42(12):2536–2544.

Bandler, J. W., Biernacki, R. M., Chen, S. H., Hemmers, R. H., and Madsen, K. (1995). Electromagnetic optimization exploiting aggressive space mapping. *IEEE Transactions on Microwave Theory and Techniques*, 43(12):2874–2882.

Bandler, J. W., Georgieva, N., Ismail, M. A., Rayas-Sánchez, J. E., and Zhang, Q.-J. (2001). A generalized space-mapping tableau approach to device modeling. *IEEE Transactions on Microwave Theory and Techniques*, 49(1):67–79.

Buzacott, J. A. and Shanthikumar, J. G. (1993). *Stochastic Models of Manufacturing Systems.* Prentice Hall.

Chang, K. J., Haftka, R. T., Giles, G. L., and Kao, I.-J. (1993). Sensitivity-based scaling for approximating structural response. *Journal of Aircraft*, 30(2):283–288.

Chen, R., Xu, J., Zhang, S., Chen, C. H., and Lee, L. H. (2015). An effective learning procedure for multi-fidelity simulation optimization with ordinal transformation. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 702–707. IEEE.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova*, 4(5):613–617.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1996). Support vector regression machines. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in neural information processing systems 9 (NIPS 1996)*, pages 155–161.

Gano, S. E., Renaud, J. E., and Sanders, B. (2005). Hybrid variable fidelity optimization by using a kriging-based scaling function. *AIAA Journal*, 43(11):2422–2433.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering.* Prentice Hall.

Haftka, R. T. (1991). Combining global and local approximations. *AIAA Journal*, 29(9):1523–1525.

Han, Z., Zimmerman, R., and Görtz, S. (2012). Alternative cokriging method for variable-fidelity surrogate modeling. *AIAA Journal*, 50(5):1205–1210.

Han, Z. H. and Görtz, S. (2012). Hierarchical kriging model for variable-fidelity surrogate modeling. *AIAA Journal*, 50(9):1885–1896.

Han, Z. H., Görtz, S., and Zimmermann, R. (2013). Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, 25(1):177–189.

Haykin, S. S. (2009). *Neural Networks and Learning Machines.* Pearson Prentice Hall.

Helton, J. C. and Davis, F. J. (2003). Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69.

Hu, J., Zhou, Q., Jiang, P., and Xie, T. (2016). An improved hierarchical kriging for variable-fidelity surrogate modeling. In *2016 International Conference on Cybernetics, Robotics and Control (CRC)*, pages 86–90. IEEE.

Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382.

Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

Leary, S. J., Bhaskar, A., and Keane, A. J. (2003). A knowledge-based approach to response surface modelling in multifidelity optimization. *Journal of Global Optimization*, 26(3):297–319.

Lewis, R. and Nash, S. (2000). A multigrid approach to the optimization of systems governed by differential equations. In *8th Symposium on Multidisciplinary Analysis and Optimization*, page 4890.

Li, J. and Meerkov, S. M. (2009). *Production Systems Engineering.* Springer.

Lin, Z., Matta, A., Li, N., and Shanthikumar, J. G. (2016). Extended kernel regression: A multi-resolution method to combine simulation experiments with analytical methods. In Roeder, T. M. K., Frazier, P. I., Szechtman, R., Zhou, E., Huschka, T., and Chick, S. E., editors, *Proceedings of the 2016 Winter Simulation Conference (WSC)*, pages 590–601. IEEE.

Lophaven, S. N., Nielsen, H. B., and Søndergaard, J. (2002). Aspects of the matlab toolbox dace. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142.

Osorio, C. and Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems. *Operations Research*, 61(6):1333–1345.

Papadopoulos, C. T., O'Kelly, M. E., Vidalis, M. J., and Spinellis, D. (2009). *Analysis and Design of Discrete Part Production Lines*. Springer.

Robinson, T. D., Eldred, M. S., Willcox, K. E., and Haimes, R. (2008). Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *AIAA Journal*, 46(11):2814–2822.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, 4(4):409–423.

Sun, G., Li, G., Stone, M., and Li, Q. (2010). A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials. *Computational Materials Science*, 49(3):500–511.

Sun, G., Li, G., Zhou, S., Xu, W., Yang, X., and Li, Q. (2011). Multi-fidelity optimization for sheet metal forming process. *Structural and Multidisciplinary Optimization*, 44(1):111–124.

Suri, R. and Leung, Y. T. (1987). Single run optimization of a siman model for closed loop flexible assembly systems. In Grant, H., Kelton, W. D., and Thesen, A., editors, *Proceedings of the 1987 Winter Simulation Conference (WSC)*, pages 738–748, Atlanta, Georgia. Association for Computing Machinery.

Tempelmeier, H. and Kuhn, H. (1993). *Flexible Manufacturing Systems: Decision Support for Design and Operation*. John Wiley & Sons.

Wahba, G. (1990). *Spline models for observational data*, volume 59. Siam.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Monographs on Statistics & Applied Probability. Chapman and Hall/CRC.

Wang, F. and Zhang, Q.-J. (1997). Knowledge-based neural models for microwave design. *IEEE Transactions on Microwave Theory and Techniques*, 45(12):2333–2343.

Wasserman, L. (2006). *All of Nonparametric Statistics*. Springer.

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372.

Watson, P. M. and Gupta, K. C. (1996). Em-ann models for microstrip vias and interconnects in dataset circuits. *IEEE Transactions on Microwave Theory and Techniques*, 44(12):2495–2503.

Xu, J., Zhang, S., Huang, E., Chen, C. H., Lee, L. H., and Celik, N. (2016). Mo2tos: Multi-fidelity optimization with ordinal transformation and optimal sampling. *Asia-Pacific Journal of Operational Research*, 33(3):1650017.

Yamazaki, W. and Mavriplis, D. J. (2013). Derivative-enhanced variable fidelity surrogate modeling for aerodynamic functions. *AIAA Journal*, 51(1):126–137.

Zhou, Q., Shao, X., Jiang, P., Zhou, H., and Shu, L. (2015). An adaptive global variable fidelity metamodeling strategy using a support vector regression based scaling function. *Simulation Modelling Practice and Theory*, 59:18–35.

Biographies

Ziwei Lin is Ph.D. candidate at Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University. Her thesis focuses on performance evaluation and optimization of manufacturing systems based on multi-fidelity models. Her email address is linziwei@sjtu.edu.cn.

Andrea Matta is Professor of Manufacturing at Department of Mechanical Engineering at Politecnico di Milano, where he currently teaches integrated manufacturing systems and manufacturing. He is Guest Professor at School of Mechanical Engineering of Shanghai Jiao Tong University. His research area includes analysis and design of manufacturing and health care systems. He is Editor-in-Chief of Flexible Services and Manufacturing Journal. His email address is andrea.matta@polimi.it.

J. George Shanthikumar is Richard E. Dauch Chair of Manufacturing and Operations Management and Distinguished Professor of Management at Purdue University. His research interests are in integrated interdisciplinary decision making, model uncertainty and learning, production systems modeling and analysis, queueing theory, reliability, scheduling, semiconductor yield management, simulation stochastic processes, and sustainable supply chain management. His email address is shanthikumar@purdue.edu.