2016

A Kriging Method for Modeling Cycle Time-Throughput Profiles in Manufacturing

Amirmahdi Tafreshian

A Kriging Method for Modeling Cycle Time-Throughput Profiles in Manufacturing

Amirmahdi Tafreshian

Thesis submitted
to the College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science in
Industrial Engineering

Feng Yang, Ph.D., Chair
Majid Jaridi, Ph.D.
Erin Leatherman, Ph.D.

Department of Industrial and Management Systems Engineering

Morgantown, West Virginia
July 2016

Keywords: Stochastic Kriging, Gaussian process, Batch sequential design, Discrete event simulation, Semiconductor manufacturing

Abstract

A Kriging Method for Modeling Cycle Time-Throughput Profiles in Manufacturing

Amirmahdi Tafreshian

In semiconductor manufacturing, the steady-state behavior of a wafer fab system can be characterized by its cycle time-throughput profiles. These profiles quantify the relationship between the cycle time of a product and the system throughput and product mix. The objective of this work is to efficiently generate such cycle time-throughput profiles in manufacturing which can further assist decision makings in production planning.

In this research, a metamodeling approach based on Stochastic Kriging model with Qualitative factors (SKQ) has been adopted to quantify the target relationship of interest. Furthermore, a sequential experimental design procedure is developed to improve the efficiency of simulation experiments. For the initial design, a Sequential Conditional Maximin algorithm is utilized. Regarding the follow-up designs, batches of design points are determined using a Particle Swarm Optimization algorithm.

The procedure is applied to a Jackson network, as well as a scale-down wafer fab system. In both examples, the prediction performance of the SKQ model is promising. It is also shown that the SKQ model provides narrower confidence intervals compared to the Stochastic Kriging model (SK) by pooling the information of the qualitative variables.

"The mysteries of eternity are known neither to you nor me
the enigma can be read neither by you nor me
behind the veil a discourse goes on about me and you
when the veil disappears there remain neither you nor me"

Omar Khayyam

Acknowledgements

The author wishes to express his deep grattitudes to his comittee chairman, Dr. Feng Yang, Associate Professor of the Deparment of Industrial Engineering, for providng support and valuable advice throughout this research, and to his comittee members Dr. Majid Jaridi, Professor of the Department of Industrial Engineering and a great mentor of mine, and Dr. Erin Leatherman, Assistant Professor of the Department of Statistics and an excellent teacher, for their valuables suggestions and encouragemnets.

Special thanks go to my parents, Kazem and Zohreh, who have gone through a lot of pains and struggles and sacreficed so much for their childeren. Also very special thanks are due to my elder brother, M.D. Amirhossein Tafreshian who always has supported and encuraged me in my life and education.

To my great Nation

Contents

List of Figures

List of Tables

Abbreviations

CT	**C**ycle **T**ime
TH	**TH**roughput
PM	**P**roduct **M**ix
GP	**G**aussian **P**rocess model
SK	**S**tochastic **K**riging model
SKQ	**S**tochastic **K**riging model with **Q**ualitative factors
DOE	**D**esign **O**f **E**xperiment
MSE	**M**ean **S**quared **E**rror
CV	**C**oefficient of **V**ariation
ARPE	**A**bsolute **R**elative **P**rediction **E**rror
IMSE	**I**ntegrated **M**ean **S**quared **E**rror
PSO	**P**article **S**warm **O**ptimization
CI	**C**onfidence **I**nterval
SCMC	**S**equential **C**onstrained **M**onte **C**arlo

Chapter 1

INTRODUCTION

Considering the huge amount of capital invested yearly in the semiconductor industry, semiconductor manufacturers are continuously searching for new capacity planning tools to support decisions made for improvement and more profit. Prior to making such decisions, manufacturers need to answer what-if questions regarding different (and possibly numerous) scenarios for product mix, production targets, and capital expansion (see e.g., Yang (2010)). Computer simulation is an essential tool to tackle this issue. One can run a simulation model before constructing or modifying a manufacturing system and predict the system's performance. Moreover, computer simulation can be utilized to specify the required capacity of each system's server to optimize the output's performance. Compared to experimenting with the physical system (when it is practical), computer simulation has been proved to be faster and more cost efficient. Schömig and Fowler (2000) introduce the semiconductor industry as an example of such systems where manufactures spend large amount of money and resources to design simulation models that mimic the behavior of real wafer fab systems. In contrast, some researchers use queueing theory to model the characteristics involved in the semiconductor industry (see e.g., Hopp et al. (2002)). Although being mathematically tractable, these models fail to consider many details of a real fab system (Jacobs et al. (2004); Wu et al. (2007)). Nevertheless, computer experiments in and of themselves are not suitable for answering what-if questions, since it may take many hours or days

to implement a single run. For this reason, we integrate computer simulation and statistical modeling in this study to analyze complex manufacturing systems. In the next section, we define our problem more precisely and define some notations that will be used throughout the paper.

1.1 Statement Of The Problem

In this research, the expected steady-state cycle time of each product is of our primary interest. Cycle time (CT) is defined as the total time it takes for a single item to traverse a pre-specified production line and become a finished manufacturing product (Hopp and Spearman (2001)). The expected CT can be characterized as a function of throughput (TH), product mix (PM), and product type. A wafer fab system can be viewed as a multi-product queueing network with K distinct products and M different stations. Each of these K products need to traverse a pre-specified sequence of M stations. We use the notation of Yang et al. (2011) as follows:

- $\{s_j, j = 1, 2, \cdots, M\}$: the number of parallel servers at station j.
- $\{u_{kj}, k = 1, 2, \cdots, K; j = 1, 2, \cdots, M\}$: the effective service rate of each resource at station j for products of type k.
- $\{\delta_{kj}, k = 1, 2, \cdots, K; j = 1, 2, \cdots, M\}$: the number of times product of type k visits station j.
- λ: the overall release rate of all the products into the system.
- $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_K)$: the product mix vector where α_k represents the share of product type k in the flow such that $\sum_{k=1}^{K} \alpha_k = 1$ and $\alpha_k \in [0, 1]$.
- $\lambda_k = \lambda\alpha_k$: the release rate of product type k to the system
- $\rho_{kj} = {}^{\delta_{kj}}/_{(s_j u_{kj})}$: the utilization of station j by product type k

In order to perform capacity/bottleneck analysis, one can take the following steps suggested by preliminary queueing analysis:

Calculate the utilization of station j, ρ_j:

$$\rho_j = \lambda \sum_{k=1}^{K} \alpha_k \rho_{kj} \ .$$

Then, find the utilization of system, denoted as x:

$$x = \min_j \rho_j \ .$$

The bottleneck station (BN) will be specified as below:

$$j_{BN} = \operatorname*{argmin}_j \rho_j = \operatorname*{argmin}_j \lambda \sum_{k=1}^{K} \alpha_k \rho_{kj} \ . \tag{1.1}$$

Equation (1.1) simply states that the BN station is the one that has the highest utilization among all M stations. This equation also implies that the BN station is a function of PM and may change with respect to different product mix vectors.

Further, we need to compute the stability constraint for our queueing network as shown below:

$$x = \lambda \sum_{k=1}^{K} \alpha_k \rho_{kj_{BN}} < 1 \ .$$

By using this constraint, we are able to obtain the system capacity, $u^*(\boldsymbol{\alpha})$, which gives us the upper limit on λ for stability of the system:

$$\lambda < \frac{1}{\sum_{k=1}^{K} \alpha_k \rho_{kj_{BN}}} = u^*(\boldsymbol{\alpha}) \ . \tag{1.2}$$

We need to utilize Equations (1.1) and (1.2) prior to modeling the CT-TH-PM surfaces, and both of these equations rely on the estimation of the effective service rate at each station for each product type, u_{kj}. There are various methods to estimate the effective service rates of the real manufacturing systems (see e.g., Hopp et al. (2002)). Herein, we trust the existing queueing methods and approximations for estimating the effective service rates and performing the preliminary queueing analysis.

Now, we discuss the formulation of CT-TH-PM surface. After applying capacity/BN analysis to normalize the overall throughput, λ, one can run simulation experiments at certain design points and collect the data required for the fitting of the CT-TH-PM surfaces. We will talk more about how we can find these design points in the Methodology section. As mentioned before, our response is the mean steady state CT of products and our independent variables are the overall product flow through the system, λ, the product mix vector, $\boldsymbol{\alpha}$, and the product type, k. As suggested by Yang et al. (2011), we invoke the following transformation to normalize the system throughput over the product mix region and estimate the expected CT as a function of x instead of λ, where $x \in [0, 1)$ and it is independent of PM:

$$x = \frac{\lambda}{u^*(\boldsymbol{\alpha})} \, . \tag{1.3}$$

Additionally, we use first $(K-1)$ α_k's to make the product mix variables independent of each other. As a result, we are interested in fitting a simulation-based model to estimate the long-run CT of products as a function of the vector of the independent variables $(x, \alpha_1, \alpha_2, \cdots, \alpha_{K-1}, k)$. In the next two sections, we state the goal of this work and the research approach to achieve these goals.

1.2 Research Objectives

The main objective of this research is to quantify for multi-product semiconductor manufacturing systems the functional dependence of the mean of steady-state CT [1] upon the input decision variables $(x, \alpha_1, \alpha_2, \cdots, \alpha_{K-1}, k)$.

[1] In the remainder of this work, we will use CT to refer to the mean of steady-state cycle times.

1.3 Research Approach

CT-TH-PM response surfaces are complex, and we chose a Gaussian Process model to fit the target surfaces because of its flexibility and ability to provide valid statistical inference. To efficiently estimate the CT-TH-PM relationships, a sequential procedure is developed to collect simulation data in batches.

The remainder of this thesis is organized as follows. Chapter 2 provides a brief review of the existing literatures. Chapter 3 describes the adopted GP model, and presents the sequential experimental design procedure. The metamodeling methods are applied to two illustrative examples and the results are given in Chapter 4. Finally, the conclusions and recommendations for further study are provided in Chapter 5.

Chapter 2

LITERATURE REVIEW

Our objective, as noted earlier, is to obtain a model for generating the mean cycle time of different products in a wafer fab as a function of system's utilization, product mix, and product type. In the literature, there exist studies devoted to the generation of such CT–TH–PM surfaces. In general, we can divide these studies in two major categories: Analytical approaches and simulation-based approaches, and each of these approaches have advantages and disadvantages. Next, we will explain some of these studies in more detail.

In analytical approaches, one may consider the wafer fab as a queueing network and thus apply queueing theory to compute cycle time in steady states by using the information of the arrival process and the service process. For instance, Jackson (1963) introduced a simple queueing network for job shop problems where the inter-arrival time for different products and the process time of different tools follow an exponential distribution and each type of product traverses a specific route of tools. It can be shown that the exact cycle time of each product in the Jackson network system can be obtained by exploiting Little's law and stationary equilibrium. Kuehn (1979) developed an approximation method, called decomposition method, which decomposes the queueing network into subsystems, and thus this method allows analysis of the queueing networks with inter-arrival times and service times as the renewal processes. Shanthikumar and Buzacott (1981) extended the Jackson network model by allowing service times having a

general distribution and applying the decomposition method introduced by Kuehn (1979). Furthermore, Whitt (1983) developed a software package called Queueing Network Analyzer which uses the decomposition method along with an approximation method to separate nodes in a job shop queueing network and analyze these nodes independently.

As noted by Shanthikumar et al. (2007), modeling and analysis of queueing systems in semiconductor manufacturing is rather complicated because it involves many tools with different configurations and processing requirements which may require more sophisticated models. Chen et al. (1988) first considered an ideal fab with a simple queueing system and applied queueing network models to obtain the cycle time of entities. Connors et al. (1996) improved the queuening network model of wafer fabs further by allowing tool groups into the model and refining the characterization of rework and scrap. Hopp et al. (2002) introduced an optimized queueing network (OQNet) system for capacity planning of new and reconfigured semiconductor manufacturing facilities. They considered a variety of common assumptions in a wafer fab such as batch processes, re-entrant flows, multiple product classes, and machine setups, and they optimized the facility cost with respect to some constraints on cycle times. The authors claimed that the results obtained by the OQNet system are not more than 30% off the simulated results. Further improvements on analytical approaches have been accomplished by researchers in recent years (see e.g., Shanthikumar et al. (2007) for further information). Although being mathematically tractable, the analytical approaches fall short in using all aspects of a real wafer fab facility because one may need several restrictions to obtain a model in closed form. Most of the time, these models tend to overestimate the cycle times since they are not flexible for different policies of handling WIPs (see e.g. Miltenburg et al. (2002)).

As an alternative to queueing network models, computer simulation has been a more flexible tool to design and analyze manufacturing systems since it allows more details of the process to be taken under consideration. Simulation models can be utilized either before a new system is created or after it has been employed when it needs substantial changes (see, e.g., Schömig and Fowler (2000)).

In the literature, one can find a lot of studies of the application of computer simulation in semiconductor manufacturing. Hung and Leachman (1996) proposed an iterative computer simulation and linear programming optimization to obtain the future cycle times as a function of product mix and work load. Sivakumar (1999) applied an online simulation-based system to optimize the cycle time and utilization of a semiconductor manufacturing facility. Park et al. (2002) proposed a simulation-based method to efficiently generate the CT-TH curves in manufacturing by exploiting a sequential simulation experiment based on a nonlinear D-optimal design. In spite of fidelity and flexibility of simulation models, Fowler and Rose (2004) claimed that it may take a long time to run a single replication for more complex manufacturing systems, and hence it would not be practical in many cases. Moreover, simulation merely provides an estimate at each single point and one may need to run several replications to improve the estimation at each single point.

With this in mind, Yang et al. (2007) developed a metamodeling approach that alleviates the major shortcomings of queueing networks and computer simulation in generation of CT–TH profiles. A metamodel is a mathematical equation in the form of polynomial regressions, splines, etc., that quantifies the results obtained by the simulation. For more information about metamodeling techniques, one can refer to Henderson and Nelson (2006). Yang et al. (2008) proposed the generalized Gamma distribution as the underlying distribution of (CT-TH) percentile curves. In this research, she introduced another metamodel to find the first three moments of CT-TH percentile curves, and thus estimated the parameters of the underlying distribution by matching the percentiles. Both of these nonlinear regression metamodels are suitable for the two-dimensional CT-TH curves. To incorporate PM as a decision variable in multi-product environment, Yang (2010) developed a neural network (NN)-based metamodeling approach. In this method, she does not treat the NN as a black box, and instead, she specifies a predetermined model for fitting based on her experience with the behavior of the response surface. Moreover, she proposed a progressive fitting approach to construct an effective and efficient network by optimizing the number of layers.

This work intends to address the same problem as in Yang (2010) by developing a GP-based metamodeling method. Compared to the NN modeling in Yang (2010), the GP method has the following advantages.

1. The GP is able to model together the CT-TH-PM profiles for all product types. This way, we can exploit the information sharing between our different products and obtain more reliable predictions.
2. Since our model is capable of handling non-smooth continuous regions, our model is fit over the entire region of inputs whereas Yang (2010) is fitting different models for each sub-region with a similar bottleneck station.
3. We do not need to assume a constant variance throughout the region of inputs which shows the capability of our proposed model in adapting to the real problems.

Chapter 3

METHODOLOGY

As mentioned in the previous chapter, cycle time of a product, the response, can be explained by its product mix and system utilization, the independent variables. These variables are considered quantitative factors and are noted by the vector $\mathbf{x}$ in our model. However, the Gaussian process model proposed here is capable of handling both quantitative and qualitative factors. Having this in our mind, we define the product type as a qualitative factor, noted by $\mathbf{z}$ with Q levels $\{c_q; q = 1, \cdots, Q\}$, to be included in our model. Therefore, the experimental design point can be shown by vector $\mathbf{w} = (\mathbf{x}^\top, \mathbf{z}^\top)^\top$ and following this notation, the random cycle time for an experimental design can be generally written as

$$\mathscr{Y}(\mathbf{w}) = \mathbf{E}\left[\mathscr{Y}(\mathbf{w})\right] + \varepsilon(\mathbf{w}) = \mathsf{Y}(\mathbf{w}) + \varepsilon(\mathbf{w}) \tag{3.1}$$

where $\mathbf{E}\left[\mathscr{Y}(\mathbf{w})\right] = \mathsf{Y}(\mathbf{w})$ is the true expected cycle time and $\varepsilon(\mathbf{w})$ is the mean zero random error which gives the response a stochastic behavior. Prior to using a GP model, we perform a simulation-based experiment of I different design points and collect the data denoted as

$$\{(\mathbf{w}_i, \mathscr{Y}_l(\mathbf{w}_i)); i = 1, \cdots, I; l = 1, 2, \cdots, n(\mathbf{w}_i)\} \tag{3.2}$$

where $\mathbf{w}_i$ represents the i^{th} experimental design point, $\mathscr{Y}_l(\mathbf{w}_i)$ is the observed response from the l^{th} replication at $\mathbf{w}_i$, and $n(\mathbf{w}_i)$ denotes the number of replications

at $\mathbf{w}_i$. Then, we apply a GP method to model the CT-TH-PM surfaces of different products together. The structure of proposed GP model has been detailed in next section.

3.1 The Gaussian Process Model

The Gaussian process (GP) model has been introduced by Sacks et al. (1989b) to fit the data from a deterministic computer experiment. This GP model is shown as

$$\mathrm{y} = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + \mathrm{Z}(\mathbf{x}) \; . \tag{3.3}$$

In (3.3), $\mathbf{x}$ is a vector of continuous factors in $\mathbb{R}^d$, $\mathbf{f}(\mathbf{x})$ is a $p \times 1$ vector of known functions, $\boldsymbol{\beta}$ is the corresponding vector of coefficients, $\mathrm{Z}(\mathbf{x})$ is a Gaussian stochastic process with mean zero and correlation matrix $\mathbf{R}(\boldsymbol{\theta})$ and the elements of $\mathbf{R}(\boldsymbol{\theta})$ are the correlations between the responses at two design points. It is worth mentioning that the constant mean β_0 is sufficient for most applications, e.g. our problem, and $\mathrm{Z}(\mathbf{x})$ is a function that maps $\mathbb{R}^d \to \mathbb{R}$. The GP model is generally a spatial correlation model, because the correlation of the response between two distinct observations becomes smaller when the design factors get farther away from each other in space. As noted by Montgomery (2008), the GP model, which provides an exact fit to the observations from the experiment, is one of the most popular models among researchers not only because of the 'exact fit' but also because of the small number of parameters involved in the model for handling so-called complex surfaces. However, this model can not be applied to the experiments with qualitative factors. Thus, Qian et al. (2008) developed a new model for deterministic computer experiments with a valid correlation function to tackle this issue. Although powerful and effective in many problems, these two models are not capable of handling the intrinsic uncertainty inherent in a stochastic computer experiment and this fact motivated Ankenman et al. (2010) to propose a new model, called the Stochastic Kriging (SK) method. In this model, which accounts for both the intrinsic and extrinsic uncertainty of the response surface,

the output of the computer experiment on the l^{th} replication at design point $\mathbf{x}$ can be shown as

$$\mathscr{Y}_l(\mathbf{x}) = \mathsf{Y}(\mathbf{x}) + \varepsilon_l(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + \mathsf{M}(\mathbf{x}) + \varepsilon_l(\mathbf{x}) \ , \tag{3.4}$$

where $\mathsf{Y}(\mathbf{x})$ is the true response at design point $\mathbf{x}$, $\mathsf{M}(\mathbf{x})$ has the similar definition to $\mathsf{Z}(\mathbf{x})$ in (3.3), and $\varepsilon_l(\mathbf{x})$ represents the independent and identically distributed random errors with mean zero and accounts for variability in the response from one replication to the other at design point $\mathbf{x}$. Finally, Wang et al. (2014) took advantage of the last two models and introduced a quite powerful and flexible method, the Stochastic Kriging model with Qualitative factors (SKQ). This model, which is suitable for our problem, has the following structure on the l^{th} replication of computer simulation at design setting $\mathbf{w}$:

$$\mathscr{Y}_l(\mathbf{w}) = \mathsf{Y}(\mathbf{w}) + \varepsilon_l(\mathbf{w}) = \mathbf{f}(\mathbf{w})^\top \boldsymbol{\beta} + \mathsf{M}(\mathbf{w}) + \varepsilon_l(\mathbf{w}) \ . \tag{3.5}$$

In (3.5), $\mathbf{w} = (\mathbf{x}^\top, \mathbf{z}^\top)^\top$ is an experimental design setting, including d continuous factors $\mathbf{x} = (x_1, x_2, \cdots, x_d)^\top \in \mathbb{R}^d$ and J qualitative factors $\mathbf{z} = (z_1, z_2, \cdots, z_J)^\top$ with each z_j having m_j levels. The polynomial term $\mathbf{f}(\mathbf{w})^\top \boldsymbol{\beta}$ has the same definition as in (3.3). We reduce the polynomial term to a constant mean, because Steinberg and Bursztyn (2004) showed that the correlation function performs very well in terms of capturing linear and quadratic trends and first-order interactions, if present, and there is no need to include any polynomial terms into the GP model unless there is sufficient evidence to infer otherwise. The terms $\mathsf{M}(\mathbf{w})$ and $\varepsilon_l(\mathbf{w})$ express the extrinsic and intrinsic uncertainties in the response, respectively, and we elaborate on them in next sections.

In order to build an SKQ model, we perform an experiment with n_i simulation replications at $\mathbf{w}_i, i = 1, 2, \cdots, k$, for k distinct design settings in total. It is worth noting that the number of replications at each design setting, n_i, can vary. According to Wang et al. (2014), the sample average of our simulation outputs at

design setting $\mathbf{w}_i$ can be exhibited as

$$\bar{\mathscr{Y}}(\mathbf{w}_i) = \frac{1}{n_i}\sum_{l=1}^{n_i}\mathscr{Y}_l(\mathbf{w}_i) = \mathbf{f}(\mathbf{w}_i)^\top\boldsymbol{\beta} + \mathsf{M}(\mathbf{w}_i) + \frac{1}{n_i}\sum_{l=1}^{n_i}\varepsilon_l(\mathbf{w}_i)\ , \tag{3.6}$$

and the $k \times 1$ vector of averaged simulation outputs is $\bar{\mathscr{Y}} = \big(\bar{\mathscr{Y}}(\mathbf{w}_1), \bar{\mathscr{Y}}(\mathbf{w}_2), \cdots, \bar{\mathscr{Y}}(\mathbf{w}_k)\big)^\top$. Moreover, the $k \times 1$ vector of averaged simulation errors is denoted by $\varepsilon = \big(\bar{\varepsilon}(\mathbf{w}_1), \bar{\varepsilon}(\mathbf{w}_2), \cdots, \bar{\varepsilon}(\mathbf{w}_k)\big)^\top$, where

$$\bar{\varepsilon}(\mathbf{w}_i) = \frac{1}{n_i}\sum_{l=1}^{n_i}\varepsilon_l(\mathbf{w}_i)\ , \quad i = 1, 2, \cdots, k\ .$$

3.1.1 The Extrinsic Variance Structure for SKQ

$\mathsf{M}(\mathbf{w})$ is a stationary Gaussian process with mean zero and spatial variance-covariance matrix denoted by Σ_{M}, where $\Sigma_{\mathsf{M}}(\mathbf{w}, \mathbf{w}') = \mathrm{Cov}\big[\mathsf{M}(\mathbf{w}), \mathsf{M}(\mathbf{w}')\big] = \mathrm{Cov}\big[\mathsf{Y}(\mathbf{w}), \mathsf{Y}(\mathbf{w}')\big]$. For k design points, Σ_{M} is a $k \times k$ matrix where the ij^{th} entry identifies the spatial covariance between the response at the i^{th} and j^{th} design settings. Based on the framework proposed by Qian et al. (2008), Wang et al. (2014) suggest the following structure for the elements of this matrix in SKQ:

$$\Sigma_{\mathsf{M}}(\mathbf{w}, \mathbf{w}') = \tau^2\left[\prod_{j=1}^{J}\varsigma_{j,z_j,z'_j}\right]\cdot K(\mathbf{x}, \mathbf{x}')\ , \tag{3.7}$$

where $\tau^2 > 0$ is the constant extrinsic variance, $\mathbf{w}_i$ and $\mathbf{w}$ are two distinct design settings, z_j and z'_j are the corresponding setting for the j^{th} qualitative factor at $\mathbf{w}$ and $\mathbf{w}'$, $\mathbf{x}$ and $\mathbf{x}'$ are the corresponding continuous factor setting at $\mathbf{w}$ and $\mathbf{w}'$, ς_{j,z_j,z'_j} is the multiplicative correlation function for qualitative variables, and finally $K(\mathbf{x}, \mathbf{x}')$ is the correlation function for continuous variables.

In the literature, there is a wide range of valid correlation functions for continuous variables and Santner et al. (2013) and Qian et al. (2008) explain some of these functions in detail. As an example, the family of exponential correlation functions is:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left\{\sum_{m=1}^{d} -\theta_m|x_m - x'_m|^p\right\}\ . \tag{3.8}$$

In (3.8), $\mathbf{x}_i$ and $\mathbf{x}_h$ are two distinct continuous factor settings in $\mathbb{R}^d$, $\theta_m > 0$ is the roughness parameter to quantify the smoothness of the response surface in the direction of coordinate $m, m = 1, 2, \cdots, d$, and parameter $p \in (0, 2]$ is a real number. Setting parameter p equal to 2 makes this correlation function infinitely differentiable and we refer to this function as the Gaussian correlation function (Ramussen and Williams, 2006). This Gaussian correlation function is not able to quantify the correlation between qualitative factors and thus Qian et al. (2008) came up with the equation in (3.7) to tackle this issue. As noted by Qian et al. (2008), we can think of $\varsigma_{j,z_j,z\prime_j}$ as a measure of similarity in two design settings that have the same values for all quantitative and qualitative factors except the qualitative factor j; i.e., in our problem, ς_{j,z_j,z'_j} explains the similarity between the response surface of two different products in any fixed product mix and throughput. For building a structure for the correlation function of the qualitative factors, Qian et al. (2008) proposed the Isotropic (or exchangeable) correlation functions (EC) which is written as

$$\varsigma_{j,z_j,z'_j} = \exp\left\{-\phi_j I[z_j \neq z'_j]\right\} , \tag{3.9}$$

where $\phi_j > 0$ is the correlation parameter for the qualitative factor j and $I[\cdot]$ is an indicator function that is equal to 1 if the expression inside the bracket holds and 0 otherwise. Notice that EC does not distinguish between different levels of a qualitative factor. Although quite simple and popular, EC can not explain the possible negative correlations between qualitative factors. Zhou et al. (2011) introduced an unrestricted correlation function (UC) to address this issue. There are two formulations for the UC function: the general formulation and the product formulation. In the general formulation, we consider all possible level combinations of the qualitative factors in $\mathbf{z}$, which has $m = \prod_{j=1}^{J} m_j$ different levels, and consequently we need to estimate $m(m-1)/2$ parameters for the qualitative factors. This aggravates the estimation problem if m is relatively large. On the other hand, there is a product formulation which copes with each factor separately and then multiplies the correlations of all qualitative factors, and thus substantially lessens the number of parameters needing to be estimated. Since we have only

one qualitative factor in our problem, there is no difference between these two formulations. Here we discuss the latter formulation and refer interested readers to Zhou et al. (2011) for more information. Zhou et al. (2011) used the hypersphere decomposition (see also, Chen et al. (2013); Rebonato and Jäckel (2011)) to build the $m_j \times m_j$ positive definite with unit diagonal elements (PDUDE) correlation matrix $\mathbf{T}_j = [\tau_{r,s}], r, s = 1, 2, \cdots, m_j$ in the following 2 steps.

Step 1. By using a Cholesky decomposition, we calculate the lower triangular matrix with strictly positive diagonal elements $\mathbf{L}$ where $\mathbf{T} = \mathbf{L}\mathbf{L}^\top$.

Step 2. For each row vector $(l_{r,1}, l_{r,2}, \cdots, l_{r,r})$ in $\mathbf{L}$ assuming that $l_{1,1} = 1$:

$$\begin{cases} l_{r,1} = \cos(\phi_{r,1}), & \\ l_{r,s} = \sin(\phi_{r,1}) \cdots \sin(\phi_{r,s-1}) \sin(\phi_{r,s}), & \text{for} \quad s = 2, \cdots, r-1 \\ l_{r,r} = \sin(\phi_{r,1}) \cdots \sin(\phi_{r,r-2}) \sin(\phi_{r,r-1}), & \end{cases}$$

where $\phi_{r,s}$ belongs to the parameter set $\Phi = \{\phi_{r,s} \in (0, \pi), s = 1, 2, \cdots, r-1; r = 1, 2, \cdots, m\}$. Note that $\phi_{r,s} \in (0, \pi)$ may produce some negative elements in the matrix $\mathbf{T}_j$; i.e., UC is able to handle both positive and negative correlation.

3.1.2 The Intrinsic Variance Structure for SKQ

So far in Section 3.1.1, we have only characterized the structure of a GP model that does not include the so-called nugget effect (see e.g., Cressie (2015)) and thus it is only suitable for modeling deterministic computer experiments. Adding the term $\varepsilon(\mathbf{w})$, which is called the intrinsic uncertainty by Ankenman et al. (2010), to the current model allows it to be applied to the outputs of a stochastic simulation experiment. Since we are only concerned about predicting the average response (mean cycle time) in our problem, we elaborate on the details of the averaged simulation error vector $\bar{\varepsilon}$ in this section. It is assumed that $\bar{\varepsilon}$ is coming from a multivariate Gaussian distribution with mean zero and the variance-covariance matrix Σ_ε. Furthermore, we assume that the averaged random simulation errors are independent and identically distributed across the k design settings since we

are not using common random numbers (CRN) through our simulation implementation (see e.g., Kelton and Law (2000)). Therefore, Σ_ε is a $k \times k$ diagonal matrix as shown below:

$$\Sigma_\varepsilon = \begin{pmatrix} \frac{\mathrm{Var}\left[\varepsilon(\mathbf{w}_1)\right]}{n(\mathbf{w}_1)} & 0 & \cdots & 0 \\ 0 & \frac{\mathrm{Var}\left[\varepsilon(\mathbf{w}_2)\right]}{n(\mathbf{w}_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\mathrm{Var}\left[\varepsilon(\mathbf{w}_k)\right]}{n(\mathbf{w}_k)} \end{pmatrix}. \tag{3.10}$$

Note that the variances of average simulation outputs can differ and this fact makes SKQ a flexible and realistic tool for our specific application. In our study, the simulation run for all products at a PM and TH are implemented at once and it is very unusual to assume that different products have the same variability in their cycle time. If there were no such flexibility in our modeling tool, we would have to perform simulation runs for each product at a specific PM and TH separately, which is significantly time-consuming.

3.1.3 Estimation and Prediction for SKQ

Based on the structure of the SQK model, we have the following list of parameters that need to be estimated:

- β_0: The constant mean
- $\boldsymbol{\theta}$: The vector of correlation parameters corresponding to d quantitative factors
- $\boldsymbol{\Phi}$: The vector of correlation parameters corresponding to j qualitative factors
- $\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\Phi})$: The correlation function between the design settings
- τ^2: The extrinsic spatial variance
- Σ_ε: The diagonal intrinsic variance-covariance matrix

Furthermore, we assume that the vector of averaged simulation outputs $\bar{\mathscr{Y}}$ has a multivariate Gaussian distribution with constant mean β_0 and variance-covarance

matrix $\Sigma = \Sigma_{\mathsf{M}} + \Sigma_{\varepsilon} = \tau^2\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\Phi}) + \Sigma_{\varepsilon}$. Following this assumption, we use the maximum likelihood estimation (MLE) method for maximizing the following log-likelihood function:

$$\ln \mathscr{L} = -\frac{1}{2}\left(k\ln(2\pi) + \ln(|\Sigma|) + \widetilde{\mathscr{Y}}^{\top}\Sigma^{-1}\widetilde{\mathscr{Y}} \right) , \tag{3.11}$$

where $\widetilde{\mathscr{Y}} = \bar{\mathscr{Y}} - \beta_0\mathbf{1}_k$, $\mathbf{1}_k$ is a $k \times 1$ vector of ones, and $|\Sigma|$ is the determinant of Σ. Ankenman et al. (2010) suggest that the diagonal elements of intrinsic variance-covariance matrix can be estimated, independent of other parameters, as follows:

$$\widehat{\Sigma}_{\varepsilon_{i,i}} = \frac{1}{n_i(n_i-1)} \sum_{l=1}^{n_i} \left(\mathscr{Y}_l(\mathbf{w}_i) - \bar{\mathscr{Y}}(\mathbf{w}_i) \right)^2, \quad i = 1, 2, \cdots, k \, . \tag{3.12}$$

After substituting $\widehat{\Sigma}_{\varepsilon}$ in (3.11), we take the following steps, suggested by Qian et al. (2008), to obtain the MLEs of other parameters.

β-**step**: Given τ^2, $\boldsymbol{\theta}$, and $\boldsymbol{\Phi}$, we can obtain $\widehat{\beta}_0$ as

$$\widehat{\beta}_0(\tau^2, \boldsymbol{\theta}, \boldsymbol{\Phi}) = \left(\mathbf{1}_k^{\top}[\tau^2\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\Phi}) + \widehat{\Sigma}]^{-1}\mathbf{1}_k \right)^{-1} \mathbf{1}_k^{\top}[\tau^2\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\Phi}) + \widehat{\Sigma}]^{-1}\bar{\mathscr{Y}} \, . \tag{3.13}$$

$(\tau^2, \boldsymbol{\theta}, \boldsymbol{\Phi})$-**step**: Given $\widehat{\beta}_0(\tau^2, \boldsymbol{\theta}, \boldsymbol{\Phi})$, our problem reduces to the following:

$$\begin{aligned} (\tau^2, \boldsymbol{\theta}, \boldsymbol{\Phi}) = \underset{\tau^2, \boldsymbol{\theta}, \boldsymbol{\Phi}}{\operatorname{argmin}} \Big[& \ln(|\tau^2\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\Phi}) + \widehat{\Sigma}_{\varepsilon}|) + \\ & (\bar{\mathscr{Y}} - \widehat{\beta}_0\mathbf{1}_k)^{\top} \left[\tau^2\boldsymbol{\theta}, \boldsymbol{\Phi}) + \widehat{\Sigma}_{\varepsilon}| \right]^{-1} (\bar{\mathscr{Y}} - \widehat{\beta}_0\mathbf{1}_k) \Big] \, , \\ \text{subject to} \quad & \theta_i > 0, \quad i = 1, \cdots, d \, . \end{aligned} \tag{3.14}$$

This problem falls into the category of constrained nonlinear multivariate optimization problems and can be easily solved by the MATLAB fuction *fmincon*.

For predicting the expected average response at any arbitrary setting $\mathbf{w}_0$, we have the following estimator:

$$\widehat{\mathsf{Y}}(\mathbf{w}_0) = \widehat{\beta}_0 + \widehat{\Sigma}_{\mathsf{M}}(\mathbf{w}_0, \cdot)^{\top} \left[\widehat{\tau}^2\mathbf{R}(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\Phi}}) + \widehat{\Sigma}_{\varepsilon} \right]^{-1} (\bar{\mathscr{Y}} - \widehat{\beta}_0\mathbf{1}_k) \, . \tag{3.15}$$

In (3.15) $\widehat{\Sigma}_{\mathsf{M}}(\mathbf{w}_0, \cdot)$ is a $k \times 1$ vector containing the estimation of the spatial correlations between arbitrary setting $\mathbf{w}_0$ and k design settings $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k$ represented as below:

$$\widehat{\Sigma}_{\mathsf{M}}(\mathbf{w}_0, \cdot) = \widehat{\tau}^2 \mathbf{r}_0(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\Phi}}) = \widehat{\tau}^2 \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix} . \tag{3.16}$$

It is worth mentioning that Ankenman et al. (2010) referred to the predictor in (3.15) as Stochastic Kriging which is an extension of the standard Kriging estimator (see e.g., Cressie (2015)). This clarifies the reason that we call this model Stochastic Kriging with Qualitative factors. In addition, we have the following mean squared error (MSE) for SKQ which enables us to make inferences about our prediction.

$$\widehat{\mathrm{MSE}}(\mathbf{w}_0) = \widehat{\tau}^2 - \widehat{\Sigma}_{\mathsf{M}}(\mathbf{w}_0, \cdot)^\top \widehat{\Sigma}^{-1} \widehat{\Sigma}_{\mathsf{M}}(\mathbf{w}_0, \cdot) + \widehat{\eta}^\top \left(\mathbf{1}_k^\top \widehat{\Sigma}^{-1} \mathbf{1}_k\right)^{-1} \widehat{\eta} \,, \tag{3.17}$$

where $\widehat{\eta} = 1 - \mathbf{1}_k^\top \widehat{\Sigma}^{-1} \widehat{\Sigma}_{\mathsf{M}}(\mathbf{w}_0, \cdot)$. Based on these terms, we can write the two-sided $100(1-\alpha)\%$ confidence interval for the average response at any arbitrary setting $\mathbf{w}_0$ in this manner:

$$\widehat{\mathrm{Y}}(\mathbf{w}_0) \pm t_{\alpha/2}(\nu) \sqrt{\widehat{\mathrm{MSE}}(\mathbf{w}_0)} \,, \tag{3.18}$$

where $t_{\alpha/2}(\nu)$ is the $100(1-\alpha/2)^{th}$ percentile of t-distribution with degrees of freedom ν and $\nu = k - \#$ of parameters.

Like any other modeling method, understanding the characterization of SKQ is not enough to make the most of this tool in terms of effectiveness and efficiency. There are two significant factors shown in the SKQ literature (see e.g., Ankenman et al. (2010)): a decent placement of design settings, and simulation efforts in each design setting. The former plays an important role on the structure of the extrinsic variance-covarince matrix, because a relatively small distance between two design settings in space affects the condition number of the information matrix and may ruin any estimation or prediction made by that model. On the other hand, the

structure of the intrinsic variance is highly affected by the length and number of simulation runs incurred in each design setting. In the next section, we discuss the design of experiment and the simulation procedure to achieve the best performance of the SKQ model.

3.2 Procedure for Modeling the Response Surface

Following the SKQ notation detailed in the previous sections, we define the relationship between the response and independent variables in CT-TH-PM surface by

$$\mathsf{Y}(\mathbf{w}) = \beta_0 + \mathsf{M}(\mathbf{w}) + \varepsilon(\mathbf{w}) \ , \tag{3.19}$$

where the factor setting $\mathbf{w}$ consists of the values for the quantitative factors $\mathbf{x} = (x, \boldsymbol{\alpha})$ and the product type as qualitative factor $\mathbf{z}$ with m_j levels equal to the number of products. $\mathsf{Y}(\mathbf{w})$ represents the cycle time estimate of a simulation at design setting $\mathbf{w}$. Prior to describing our design of experiment (DOE), we elaborate on the computer simulation effort made at each quantitative design setting $\mathbf{x}$.

3.2.1 Computer Simulation Effort

Assuming that we have d products in our manufacturing fab and each of them has a specific sequence of machines to visit, the computer simulation runs at each design point have the following inputs and outputs:

Simulation Inputs:

- Quantitative design factor setting $\mathbf{x} = (x, \alpha_1, \alpha_2, \cdots, \alpha_{d-1})$
- Length of a simulation run specified by the total number of finished products collected in the steady-state, $Q(\mathbf{x}) = Q_1(\mathbf{x}) + Q_2(\mathbf{x}) + \cdots + Q_d(\mathbf{x})$ where $Q_i(\mathbf{x})$ is the number of finished product of type $i, i = 1, 2, \cdots, d$
- Number of simulation replications at each design point, $n(\mathbf{x})$

Simulation Outputs:

- d steady-state mean cycle time estimate for each product at design point $\mathbf{x}$ denoted by

$$\mathrm{Y}_i(\mathbf{x}) = \frac{1}{n(\mathbf{x})} \sum_{l=1}^{n(\mathbf{x})} \overline{CT}_l^i(\mathbf{x}) \ , \tag{3.20}$$

  where $\overline{CT}_l^i$ is the average cycle time for Q_i products of type i on the l^{th} replication at design point $\mathbf{x}$.

The design of experiment in the following section specifies the input design settings at which we perform simulation runs. There is no need to worry about the design of experiment for the qualitative variables in this study, because we obtain the cycle time estimate for all products at once on a simulation run and thus we use all levels of our qualitative variable, product types, at each design point $\mathbf{x}$. There are two reasons for dealing with our qualitative factor in such manner. Firstly, it is computationally more efficient to obtain design of experiment for the continuous factors only, and secondly, we take the advantage of all information acquired on each simulation run.

We use a two-step procedure (see e.g., Yang (2010)) to find the proper number of replications at each design point. In the first step, we will run the simulation for n_0 replications and collect the sample set $\{\overline{CT}_l^i, i = 1, 2, \cdots, d, l = 1, 2, \cdots, n_0\}$ with $\overline{CT}_l^i$ representing the average cycle time for product i on the j^{th} replication, computed as shown below:

$$\overline{CT}_l^i(\mathbf{x}) = \frac{1}{Q_i(\mathbf{x})} \sum_{q=1}^{Q_i(\mathbf{x})} CT_{lq}^i(\mathbf{x}) \ ,$$

where CT_{lq}^i is the cycle time of q^{th} finished product i on the j^{th} replication. Then, the initial sample mean and variance of each product will be calculated as

$$\mathrm{Y}_{i,0}(\mathbf{x}) = \frac{1}{n_0(\mathbf{x})} \sum_{l=1}^{n_0(\mathbf{x})} \overline{CT}_l^i(\mathbf{x}) \ , \tag{3.21}$$

and

$$\mathrm{Var}[\mathrm{Y}_{i,0}(\mathbf{x})] = \frac{\hat{\sigma}^2_{i0}}{n_0} = \frac{1}{n_0(Q_i(\mathbf{x})-1)} \sum_{l=1}^{Q_i(\mathbf{x})} (CT^i_{lq}(\mathbf{x}) - \overline{CT}^i_l(\mathbf{x}))^2 \,. \tag{3.22}$$

Finally, $n(\mathbf{x})$ is computed as

$$n(\mathbf{x}) = \max\left(\left\lceil \frac{\hat{\sigma}^2_{1,0}}{\sigma^2} \right\rceil, \left\lceil \frac{\hat{\sigma}^2_{2,0}}{\sigma^2} \right\rceil, \cdots, \left\lceil \frac{\hat{\sigma}^2_{d,0}}{\sigma^2} \right\rceil \right) , \tag{3.23}$$

where σ^2 is a pre-specified constant variance. In the second stage, we perform the $n(\mathbf{x}) - n_0(\mathbf{x})$ follow-up runs to obtain the cycle time estimate in (3.20). If n_0 is large enough, this method guarantees that the average mean cycle time variance at design setting $\mathbf{x}$ is less than σ^2. Note that we do not need to assume a constant variance on the simulation outputs and that is one of the prominent advantages of SKQ over previous approaches. However, this variance affects the accuracy of the simulation outputs and accordingly influences the prediction accuracy made by our model. Yang (2010) suggests to choose σ^2 small enough to ensure a high precision, say $\gamma\%$, in the simulation outputs. More precisely, she recommends to set $\sigma = 4\% \times c_{min}$ where c_{min} is a rough estimate of the smallest expected cycle time based on the user's past experience.

3.2.2 Design of Experiment

In the literature, there are two approaches for designing an experiment with a budget of $\mathcal{N}$ design points: classical DOE and sequential DOE. In classical DOE (see e.g., Montgomery (2008)), particularly in factorial designs, we introduce a few equally-spaced levels for each continuous variable and allocate all $\mathcal{N}$ design points to the combination of these levels. In sequential DOE (see e.g., Mitchell and Morris (1992)), the experiment budget will be used in a multistage procedure. It has been shown that the sequential methods outperform the classical approaches in a variety of circumstances, since models have a chance to "learn" from previous stages. The sequential approaches can be divided into two groups based on the number of design points added at each experimental stage: a fully sequential method that

adds one point at a time, and batch sequential method where a batch of n_b design points are added at each stage. In our research we use a batch sequential method since it is computationally more efficient in terms of constructing the design and it is less likely for our criterion to converge into a local optima (Loeppky et al., 2010). Figure 3.1 gives a schematic presentation of our modeling procedure summarized in a flowchart. We have already talked about the **Model Fitting** step. Other steps will be detailed in the following sections.

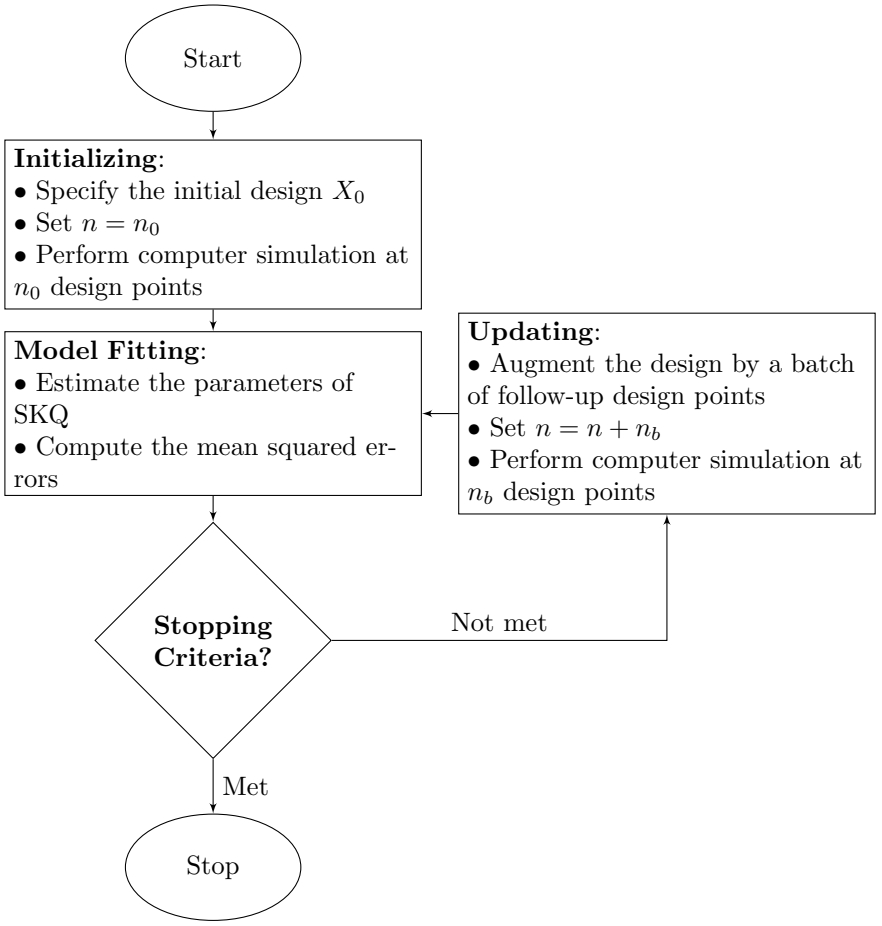


FIGURE 3.1: Model fitting procedure with a batch sequential method

3.2.2.1 Initial Design

At the first stage, we consume a portion of the fixed sampling budget for the initial design. This initial design should be large enough to let our model provide a reliable estimation of parameters, because we exploit these parameter estimates to

come up with a follow-up design at the next step. Ranjan et al. (2008) recommend allocating $25\% - 35\%$ of the sampling budget to the initial design. There is a wide tendency for researchers to use space-filling designs (see e.g., Mitchell and Morris (1992)), since they cover the input space well, and thus they can usually give a rough approximation of the response behavior. However, special care needs to be taken for our problem since the input space is not a regular hypercube and is subjected to a linear constraint. So, we are not able to use an OA-based or maximin Latin Hypercube Design (LHD). However, Golchi and Loeppky (2015) have recently proposed a novel method to obtain space-filling designs for constrained regions by proposing a Sequential Monte Carlo based algorithm to find the design points. More precisely, this approach consists of two separate algorithms: a Sequential Constrained Monte Carlo (SCMC) algorithm is used first to get a large uniform sample over the constrained input region of the continuous variables. Second, a sequential selection algorithm is used to find a space-filling design from the sample obtained in first step based on a distance-based criterion. In practice, we could not find an advantage of using the first algorithm in our research since using a large grid of points in our constrained space region can be viewed as a fairly uniform sample of points. However, we exploited the second algorithm proposed by Golchi and Loeppky (2015) with some modifications (to be revealed later) to obtain a maximin design. We explain this algorithm in detail below.

First, we obtain a uniform sample of size N (a large number) over our constrained region, denoted as $\Xi = \{\xi_1, \xi_2, \cdots, \xi_N\}$. Also, we define the distance function, $\delta(\xi_j, x_i)$ that calculates the Euclidean distance between ξ_j and x_i. Then, we start with a null set of design points, denoted as s, and we take n steps sequentially to find our initial design set of size n, $s^n = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$. For the first step, a design point is sampled randomly from Ξ, and we set $s^1 = \{\mathbf{x}_1\}$. We also calculate the Euclidean distance between $\mathbf{x}_1$ and all the design points in S, arranged in a vector of size N, denoted as ψ^1. For step i $(2 \leq i \leq n)$, we update the j^{th} element of vector ψ^i as below:

$$\psi_j^i = \min\{\psi_j^{i-1}, \delta(\xi_j, \mathbf{x}_{i-1})\} \quad j = 1, \cdots, N \tag{3.24}$$

$\mathbf{x}_i$ is found as a design point in Ξ that has the maximum distance in ψ^i, and update s^i accordingly. Note that we only need to find the distance between all the points in S and the last design point added to s at each step which saves us a huge amount of time. Golchi and Loeppky (2015) discuss that this algorithm has a better result compared to the previous methods where all points are added at once. In our research, we modify the first step to avoid extrapolation using our proposed model. In other words, in the first step we choose the centroid of the input space instead of sampling from Ξ. By this modification, the algorithm is guaranteed to choose all the extreme points in the following steps automatically.

3.2.2.2 Stopping Criteria

In our multistage procedure, after the step of constructing the SKQ model and estimating the parameters, we calculate the mean cycle times and mean squared errors over a large grid set of evaluation settings $\{\mathbf{w}_h = (\mathbf{x}_h^\top, \mathbf{z}_h^\top)^\top\}$ where $\mathbf{x}_h \in [0,1]^d$ with respect to $x_{h,1} + x_{h,2} + \cdots + x_{h,d-1} \leq 1$ and $\mathbf{z}_h$ represent the product type. This evaluation set must be large enough to ensure that the whole feasible region of interest has been covered properly. Then the estimate of coefficient of variation (CV) is calculated at each evaluation setting as

$$\widehat{\mathrm{CV}}(\mathbf{w}_h) = \frac{\sqrt{M\widehat{S}E(\mathbf{w}_h)}}{\widehat{Y}(\mathbf{w}_h)} \quad h = 1, 2, \cdots, H \tag{3.25}$$

It can be shown that $\max_h(\widehat{\mathrm{CV}}(\mathbf{w}_h))$ provides an estimate of the maximum absolute relative prediction error over this set computed as shown below:

$$\mathrm{ARPE}(\mathbf{w}_h) = \frac{|\widehat{Y}(\mathbf{w}_h) - \mathsf{Y}(\mathbf{w}_h)|}{\mathsf{Y}(\mathbf{w}_h)} \quad h = 1, 2, \cdots, H \tag{3.26}$$

where $\mathsf{Y}(\mathbf{w}_h)$ is the true average response at setting $\mathbf{w}$. Therefore we define the stopping criteria to be the fixed budget of $\mathcal{N}$ design points and the prediction precision $\delta\%$. In other words, we stop the iterative proposed procedure if we run out of the budget of $\mathcal{N}$ design points or if we achieve a maximum $\widehat{CV}(\mathbf{w}_h)$ less than $\delta\%$.

3.2.2.3 Design Augmentation

In batch sequential design, one refers to adding a batch of n_b design points to the existing set of points as the design augmentation step. There is a variety of methods in the literature for adding the follow-up design points (see e.g., Shewry and Wynn (1987)) . In most of these methods, the researcher takes advantage of the information obtained in previous steps and finds new design points to optimize a criterion function of model parameters with respect to the design factors' space. Here we discuss a few methods that have been shown to be successful in different applications; for further information we refer you to Loeppky et al. (2010).

Sacks and Schiller (1988) select a new batch design X_b that minimizes the criterion function

$$\max_{\mathbf{x}\in[0,1]^d} \widehat{MSE}(\mathbf{x}) \,, \tag{3.27}$$

with $\widehat{MSE}(\mathbf{x})$ being calculated similarly to (3.17) but only for continuous variables and the matrix $\widehat{\Sigma}_{\mathsf{M}}$ includes the entire design $X = (X_0^\top, X_b^\top)^\top$. This algorithm selects a batch of follow-up design points that minimizes the maximum estimated MSE over the input space. The Max MSE method incorporates performing several numerical optimization problems during the design optimization when the factors are continuous and this issue diminishes its popularity among different criteria.

Minimizing the Integrated MSE (IMSE) is another criterion, suggested by Sacks et al. (1989a,b), and it minimizes the following integration:

$$\int_{\mathbf{x}\in[0,1]^d} \widehat{MSE}(\mathbf{x}) d\mathbf{x} \,, \tag{3.28}$$

where again $\widehat{MSE}(\mathbf{x})$ is calculated similarly to (3.17) but only for continuous variables and the matrix $\widehat{\Sigma}_{\mathsf{M}}$ involves the whole design $X = (X_0^\top, X_b^\top)^\top$. In spite of being more computationally efficient than the Max MSE, the IMSE criterion is still time-consuming in terms of finding a new batch of design points.

Shewry and Wynn (1987) pointed out that the correlation matrix $\mathbf{R}(\theta)$ contains the amount of information in the experiment obtained by a Gaussian Process model with $\boldsymbol{\beta} = \beta_0$, and introduced the Maximum Entropy criterion to be utilized

for finding the follow-up design points. The correlation matrix $\mathbf{R}$ can be rewritten as a block matrix

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{0,0} & \mathbf{R}_{0,b} \\ \mathbf{R}_{b,0} & \mathbf{R}_{b,b} \end{pmatrix} , \tag{3.29}$$

where $\mathbf{R}_{0,0}$ indicates the correlation matrix corresponding to the previous design, $\mathbf{R}_{0,b}$ represents the correlation matrix corresponding to the design added, and $\mathbf{R}_{b,0} = \mathbf{R}_{0,b}^{\top}$ denotes the correlation matrix between the previous and added designs. In the Maximum Entropy function, the determinant of the matrix $\mathbf{R}$ is maximized, which is equivalent to maximizing the determinant of the $n_b \times n_b$ matrix

$$(\mathbf{R}_{b,b} - \mathbf{R}_{b,0}\mathbf{R}_{0,0}^{-1}\mathbf{R}_{0,b}) . \tag{3.30}$$

It is needless to say that working with the second equation is less intensive, because once one calculates the inverse of the matrix $\mathbf{R}_{0,0}$, there is no need to work with the possibly huge $n \times n$ matrix. The Maximum Entropy criterion has been shown to require less computational effort than the other two methods (Loeppky et al. (2010)). However, it still requires calculating the inverse or determinant of matrices, which may take considerable time for a programming software.

Johnson et al. (1990) proved that maximizing the entropy is equivalent to maximizing the minimum Weighted Distance, and hence proposed the Maximin Weighted Distance criterion denoted as

$$\max_{X_b \in [0,1]^2} \min_{\mathbf{x},\mathbf{x}' \in X} \sqrt{\sum_{j=1}^{d} \theta_j (x_j - x'_j)^2} , \tag{3.31}$$

where θ_j indicates the weight of the distance between two distinct design points in j^{th} dimension and is equivalent to the continuous correlation parameter in SKQ. The Maximin Weighted Distance method significantly reduces the amount of time needed for obtaining a follow-up design. Further, Loeppky et al. (2009) prove that the Maximin Weighted Distance is based on the Max MSE criterion. For finding the follow-up design in computer experiments, Loeppky et al. (2010) showed that the Maximin Weighted Distance and Maximum Entropy outperform the other methods based on the results of the maximum prediction error and the root mean

squared error criteria in some applications.

We implemented both the Maximin Weighted Distance and Maximum Entropy methods in our first case study and we found very similar results for all prediction accuracy measures except one important measure. We observed that selecting the design points using the Maximin Weighted Distance criterion will not necessarily reduce the estimated CV as the algorithm proceeds with more points. On the other hand, the follow-up design points selected using the Maximum Entropy criterion consistently reduces the estimated CV until it gets smaller than a prespecified δ. According to this experience, we choose to exploit the Maximum Entropy criterion for finding the follow-up design points. Namely, we are seeking a follow-up design that maximizes the Entropy defined in (3.30) over the entire design input region. We want the sum of $(d-1)$ mixture variables to be less than 1 ,and thus there is a nonlinear optimization involved in maximizing (3.30) subject to the linear feasibility constraint $x_{h,1} + x_{h,2} + \cdots + x_{h,d-1} \leq 1$, and thus, there are several methods to solve this optimization problem. A time-consuming but quite reliable approach is proposed by Fedorov (1972). In the Federov Exchage method, a large grid of feasible points $\mathscr{G} = \{\mathbf{x}_g, g = 1, 2, \cdots, G\}$ in $[0,1]^d$, that fairly represents the whole feasible area, is considered to be design point candidates. Initially, a starting solution X_b of size n_b from this candidate set would be generated. Next, we try all pairwise exchanges of a design point in X_b with a design point in $\mathscr{G}$ and repeat this process until we see no improvement in our objective function. Beside the Federov Exchange method, a variety of other exact or approximate optimization algorithms have been proposed to expedite the process of finding a decent optimal design. Recently, Leatherman et al. (2014) introduced the use of Particle Swarm Optimization (PSO) for designing computer experiments and it has been shown that this method provides near optimal designs in a timely manner. Next, we present a brief review of this algorithm.

The PSO algorithm starts with a large number of particles (design candidates) where these particles are scattered randomly over the input space of interest. Let's say that we are interested in finding n_b followup design points. Define N_{des} as the number of particles and rewrite each candidate matrix of size $n_b \times d$, as a

$dn_b \times 1$ vector, and therefore, we start with N_{des} vectors of size $dn_b \times 1$, denoted as $\{\vartheta_i^t\}_{i=1}^{N_{\text{des}}}$. At each iteration t, update the location of these particles by using the following equation and evaluate the objective function for the new particles:

$$\vartheta_i^{t+1} = \vartheta_i^t + v_i^{t+1} \,, \tag{3.32}$$

where ϑ_i^t and ϑ_i^{t+1} are the current and future locations of the i^{th} design, respectively. Each particle is updated by v_i^{t+1} which is defined as:

$$v_i^{t+1} = \theta v_i^t + \alpha \epsilon_{1i}^t \circ (\vartheta_i^t - g^t) + \beta \epsilon_{2i}^t \circ (\vartheta_i^t - p_i^t) \,. \tag{3.33}$$

In (3.33), g^t is the particle (design point) that gives the best global (among all design points) objective function through time t, and p_i^t is the particle that produces the best value of objective function for the i^{th} particle through time t. The symbol $\circ$ presents the element-wise product of vectors, ϵ_{1i}^t and ϵ_{2i}^t are random vectors with elements coming from Unif$[0, 1]$, α and β are weights that control the steps toward the global and particle-best locations, $\theta \in [0, 1]$ is called the "inertia" parameter and we specify a lower and an upper limit of 0.25 for the steps taken at each iteration (i.e., $v_i^t \in [-0.25, 0.25]$). This process will be continued until iteration N_{iter} and the design yielding the best criterion value is selected.

In the PSO algorithm, like in other metaheuristic methods, choosing an appropriate starting solution can improve the efficiency of finding a global optimum. Leatherman et al. (2014) suggest starting the algorithm with N_{des} particles that are selected based on an LHD design with the Maximin criterion. There are two major shortcomings in using this method in the context of our study: (1) the input space is assumed to be a hypercube in LHD designs which is not the case in our study; (2) their method is not suitable for augmenting an existing design. Therefore, we propose a starting solution based on the conditional Maximin algorithm discussed earlier for the initial design. Suppose that we currently have m design points in the input space of continuous variables and we are looking for n_b follow-up design points to add to the existing design. The algorithm in Figure 3.2 shows the steps for finding N_{des} designs (or the locations of the particles) to

start the PSO algorithm. Step 1 is the key step in our proposed method, since we calculate the minimum distance between each candidate point and the existing design points and find the weights accordingly. Using these weights in the sampling step enables us to avoid selecting the candidate points which are very close to the existing points. For each design k ($k = 1, \cdots, N_{des}$), we initialize the sequential algorithm with weighted sampling from Ξ, and find the follow-up points based on the conditional Maximin criterion as discussed earlier in the initial design section.

A Sequential Maximin Design for Initializing the PSO Algorithm

Input: $\Xi = \{\xi_1, \xi_2, \cdots, \xi_N\}$, a large grid of points over the constrained region
δ, a function for calculating the Euclidean distance between 2 points:
$s_0 = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, a sequence of the existing design points

1: find the weights of the points in Ξ
1-1: $\psi_j^0 \leftarrow \min_i \delta(\xi_j, \mathbf{x}_p)$, for $\xi_j \in \Xi$, $j = 1, \cdots, N$, $\mathbf{x}_p \in s_0$, $p = 1, \cdots, m$
1-2: $w_j \leftarrow 1 - e^{-(\psi_j^0)^2}, j = 1, \cdots, N$

2: **for** $k = 1$ to N_{des} **do**

3: Initialize the design:
3-1: sample $\mathbf{x}_1^k$ from Ξ with the weights w_i
3-2: $s_k^1 = \{\mathbf{x}_1^k\}$
3-3: $\psi_j^1 = \min\{\psi_j^0, \delta(\xi_j, \mathbf{x}_1^k)\}, j = 1, \cdots, N$

4: **for** $i = 2$ to n_b **do**
4-1: $\psi_j^i = \min\{\psi_j^{i-1}, \delta(\xi_j, \mathbf{x}_{i-1}^k)\}, j = 1, \cdots, N$
4-2: $\mathbf{x}_i^k \leftarrow \mathbf{x}_{j_{max}}$, where $j_{max} = \underset{j}{\operatorname{argmax}}\ \psi_j^i$
4-3: $s_k^i = \{\mathbf{x}_1^k, \mathbf{x}_2^k, \cdots, \mathbf{x}_i^k\}$
5: **end for**

6: **end for**

Output: $s_k^{n_b}, k = 1, \cdots, N_{des}$, N_{des} designs of size n_b

FIGURE 3.2: The procedure to find a starting solution for the PSO algorithm

In this chapter, we described the structure of the SKQ model and the procedure for the simulation and design of experiment. In the next chapter, we show the results of applying this tool to the CT-TH-PM profiles in real manufacturing systems.

Chapter 4

EMPIRICAL STUDIES

In this chapter, we evaluate our proposed Kriging method with two well-known empirical examples: a Jackson network system and a scale-down real wafer fab model. The primary intention of these studies is to survey the prediction performance of the Kriging model. This model can be utilized to estimate the CT-TH-PM profiles of manufacturing systems with any number of product types. However, in the following examples, we consider simple systems where the number of product types is three. As a result, we are able to demonstrate the division of input space based on the bottleneck regions and corresponding response surfaces. Moreover, we restrict utilization to change only from 0.75 to 0.85, since this is usually the range of utilization in which semiconductor manufacturing industries run their facilities (see e.g., Hopp (2011)). Furthermore, we set the desired prediction error $\delta\%$ to be 7%, i.e., we stop our sequential fitting process once the prediction error becomes lower than 7%. Next, we illustrate the parameters of interest in each case followed by some results.

4.1 A Jackson Network System

Our first case study belongs to a simple Jackson network system with three products ($K = 3$) and three stations ($M = 3$). As mentioned earlier, Jackson network

is a job shop problem where jobs visit a predetermined sequence of stations. Further, inter-arrival times and service times follow an exponential distribution, and there is no failure for any machine. Any Jackson network system can be completely characterized by the following sequences of parameters: the number of parallel machines at each station $\{s_j\}_{j=1}^{M}$, effective service rates for each product at each station$\{u_{kj}\}_{k=1,j=1}^{K,M}$, and the number of times each product visits each station $\{\delta_{kj}\}_{k=1,j=1}^{K,M}$. Table 4.1 clearly defines the system configuration of the Jackson network in this study.

Station 1	**Station 2**	**Station 3**
$s_1 = 1$	$s_2 = 1$	$s_3 = 1$
$u_{11} = u_{21} = u_{31} = u_1 = 4$	$u_{12} = u_{22} = u_{32} = u_2 = 3$	$u_{13} = u_{23} = u_{33} = u_3 = 2.8$
$\delta_{11} = 1$	$\delta_{12} = 3$	$\delta_{13} = 2$
$\delta_{21} = 3$	$\delta_{22} = 2$	$\delta_{23} = 1$
$\delta_{31} = 2$	$\delta_{32} = 1$	$\delta_{33} = 1$

TABLE 4.1: System configuration of a 3-product, 3-station Jackson network

Following the queuing analysis described in Chapter 1, we can divide the input space into three sub-regions with a constant bottleneck station Ω_k given the information in Table 4.1. Figure 4.1 shows the partition of input space into the sub-regions at any given level of utilization.

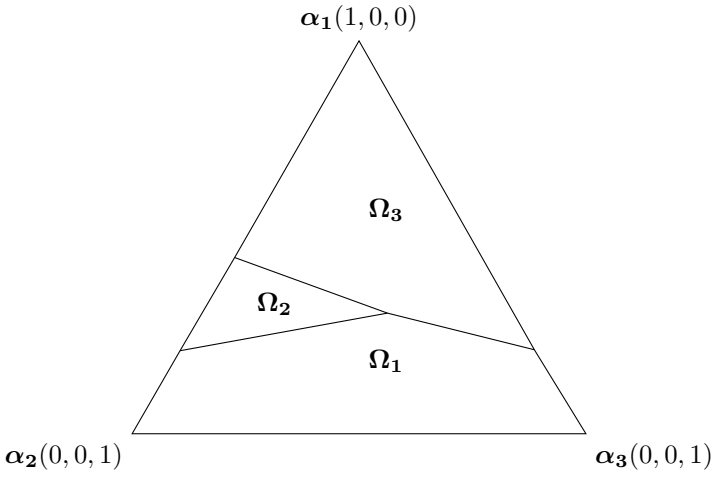


FIGURE 4.1: Partition of the input space in a Jackson network at any utilization

This figure suggests that each station serves as the bottleneck for some combinations of product mix variables. For an open Jackson network, we can obtain the true mean CT of a product as a function of PM at any level of utilization denoted as $c_k(\boldsymbol{\alpha}, x)$:

$$c_k(\boldsymbol{\alpha}, x) = \sum_{j=1}^{M} \frac{\delta_{kj}}{u_j \left[1 - x\left(\frac{\sum_{k=1}^{K} \alpha_k \delta_{kj}/u_j}{\sum_{k=1}^{K} \alpha_k \delta_{k\nu}/u_\nu}\right)\right]}, \tag{4.1}$$

where ν is the bottleneck station. Thus, we can compare the estimates provided using our proposed model with the true values.

In the preliminary queueing analysis, we do not need to specify the route of each product. However, for running the simulation emulator, we define the following sequences of the three stations as the deterministic routes for each of the three products:

- Product 1: $\{3, 1, 2, 3, 2, 3\}$
- Product 2: $\{1, 2, 1, 3, 1, 2\}$
- Product 3: $\{2, 1, 3, 1\}$

Next, we define the parameters involved in DOE of the Jackson network model.

4.1.1 Design of Experiment for a Jackson Network System

Loeppky et al., 2010 suggest that as a rule of thumb at least $10 \times d$ design points are needed for the initial design where d is the number of continuous variables in the input space. In our example, we have 3 continuous variables α_1, α_2, and x which suggests starting our fitting procedure with at least 30 design points. Based on our experience of working with the CT-TH-PM surface of the Jackson network model, we set the number of initial design points to be 50. With a space-filling design of size 50, we are able to obtain an appropriate estimate of the SKQ parameters.

In sequential designs, the number of follow-up design points is a key feature of a successful design augmentation. On the one hand, we may be trapped by a local-optimum if we pick a small number of points. On the other hand, the process of

finding an optimal design becomes computationally intensive if we choose a large number of design points. Based on our experience with the Particle Swarm Optimization algorithm, we choose the number of follow-up design points to be 25. For setting the other parameters of the design augmentation algorithm, Leatherman et al. (2014) suggest the values presented in Table 4.2.

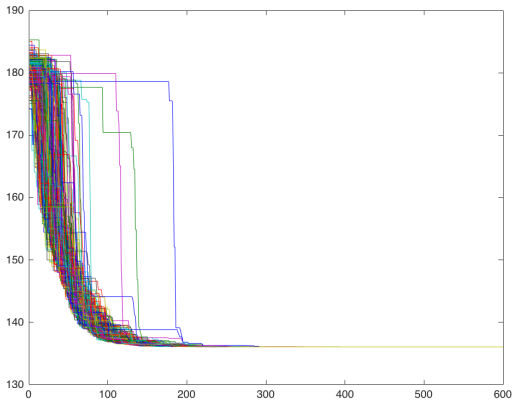
Figures 4.2a and 4.2b depict the convergence of the maximum Entropy objective function of particle-best and global locations, respectively, for adding a batch of 25 design points. Note that we are minimizing the negative Entropy in these figures which is equivalent to maximizing the Entropy. As shown in Figure 4.2, the PSO algorithm usually converges before 600 iterations, and this fact motivated us to add another criterion for stopping the PSO algorithm. We stop the algorithm if the difference between the global-best in current iteration g^t and the global-best in 50 iterations earlier g^{t-50} is less than 1×10^{-2}. Adding this criterion to our algorithm saved us a considerable amount of time for finding the follow-up designs.

θ	0.5
α	2
β	2
N_{des}	$4 \times n \times d = 300$
N_{iter}	$2 \times N_{des} = 600$

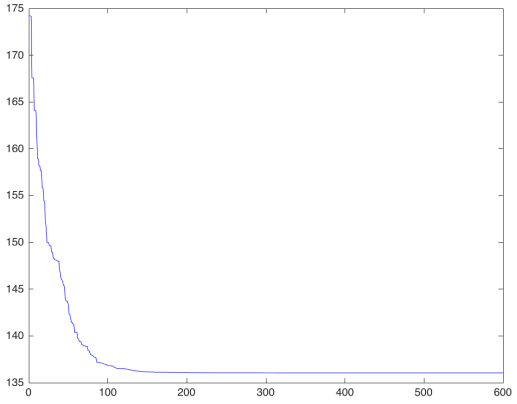
TABLE 4.2: The parameter setting of the PSO algorithm

4.1.2 Results for A Jackson Network System

In this section, we evaluate the prediction performance of our proposed model. First, we compare our estimate to the true CT over a large set of points in the input space. Second, we compare the prediction performance of the SKQ model and the SK models to show the effectiveness of pooling the information of the three products.



(A) Particle-best



(B) Global

FIGURE 4.2: The convergence of the PSO algorithm

4.1.2.1 Comparison to the True Cycle Times

As mentioned earlier, we can find the true CT by using (4.1). Therefore, we define a large set of evaluation points $\mathscr{S}_0 = \{(\alpha_1^l, \alpha_2^l, x^l), l = 1, \cdots, L\}$ including almost 100,000 evenly spaced points in the input space of continuous variables and obtain the true CT of the three products at each point. For estimating the CT's at these points, we follow the procedure in Section 3.2 and use (3.15) and (3.17) to estimate the CT's and their MSE's, respectively at each point in $\mathscr{S}_0$. It is needless to say, that we will obtain different models each time that we perform the fitting process because of not only the stochastic nature of the simulation runs, but also the several possible outcomes for the design of experiment. Thus, we repeat the fitting process of fitting for 100 macro replications, and each time we find the

relative prediction error of all the evaluation points. The relative prediction error is calculated as:

$$re = \frac{c_k(\alpha_1^l, \alpha_2^l, x^l) - \hat{c}_k(\alpha_1^l, \alpha_2^l, x^l)}{c_k(\alpha_1^l, \alpha_2^l, x^l)}, \tag{4.2}$$

where $c_k(\alpha_1^l, \alpha_2^l, x^l)$ is the true CT, and $\hat{c}_k(\alpha_1^l, \alpha_2^l, x^l)$ is the SKQ estimate of the CT of product k at point l. For each macro replication, the following statistics regarding the relative prediction errors have been calculated: min, 2.5^{th} percentile, 5^{th} percentile, 50^{th} percentile, 95^{th} percentile, 97.5^{th} percentile, and max. Then, the box plot of each statistic over 100 macro replications has been obtained and demonstrated in Figure 4.3. Figure 4.3 indicates that at least 95% of the predic-

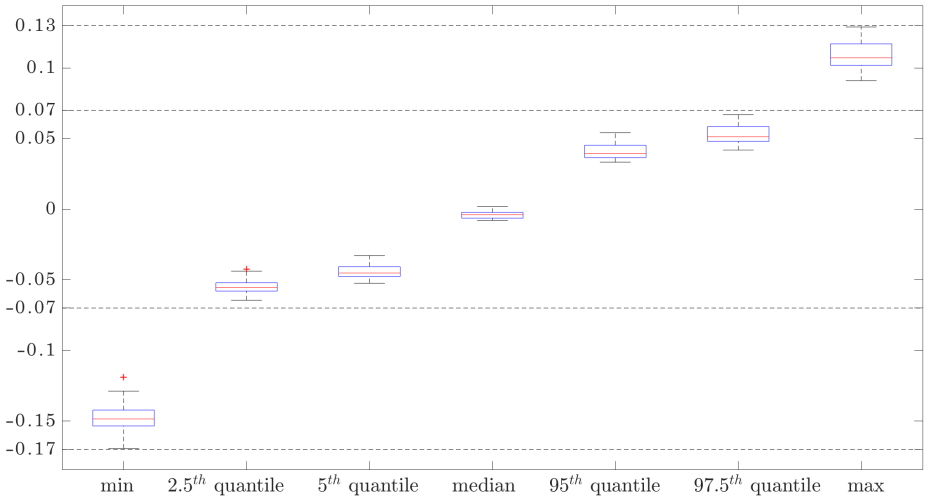


FIGURE 4.3: The box plot of different percentiles of the relative prediction errors in a Jackson network system over 100 macro replications

tion errors are less than 7%. Based on the box plot for the 50^{th} percentiles, our predictions are centered at zero and from similar frequencies in the right and left side of point zero we can infer that there is no sign of bias in our predictions. This figure also shows that in the worst case the prediction error is still between -17% and 13%. In order to investigate the prediction power of the proposed model, we choose one of the SKQ models and present its estimates and the true responses in Figure 4.4. For the sake of graphical presentation, we plot the CT of the products with respect to one mixture variable α_1 for different levels of the two other continuous variables α_2 and x. Also, the parameter estimates of this SKQ model is listed in Table 4.3.

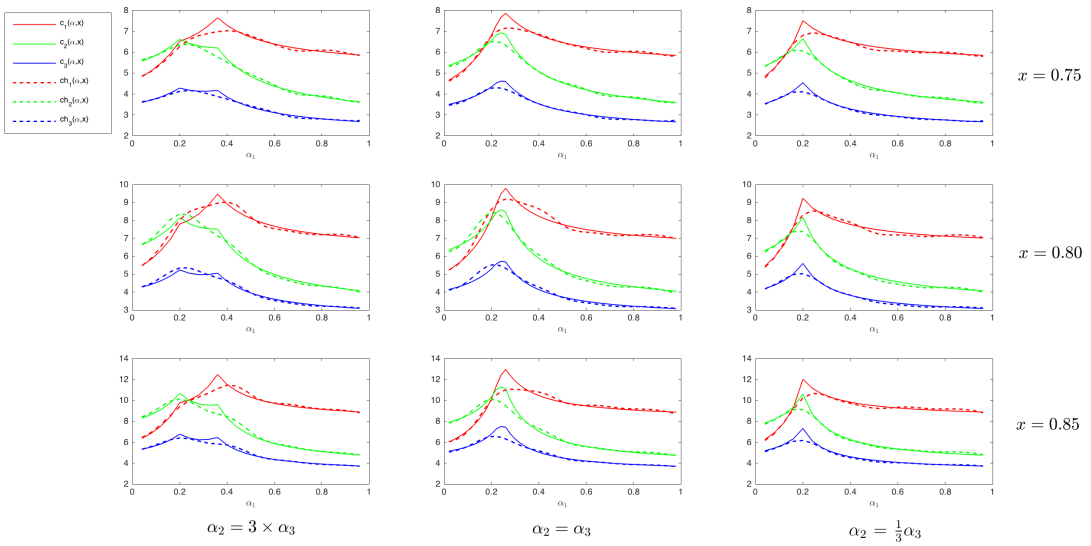


FIGURE 4.4: The true CT and its prediction for each product as a function of α_1 in 3 different levels of utilization and 3 constant ratio of α_2 and α_3

$\hat{\beta}_0$	0.4002
$\hat{\sigma}$	0.1078
$\hat{\theta}_1$	22.4578
$\hat{\theta}_2$	3.4288
$\hat{\theta}_3$	0.5771
$\hat{\phi}_1$	0.9101
$\hat{\phi}_2$	0.7313
$\hat{\phi}_3$	0.1339

TABLE 4.3: The parameter estimates of an SKQ model

As depicted in Figure 4.4, the SKQ model is able to predict the response accurately for almost every points except for those that are close to the region where the bottleneck station changes. The cycle time increases dramatically in these regions and it is hard for the SKQ model to provide accurate estimates in such points. However, the relative prediction error calculated at these points is still reasonably low, and the estimates at the other points is not affected significantly.

We can also obtain some information about the response surface by looking at the parameter estimates. It can be seen from the plots in Figure 4.4 that the response

surface is not smooth in the direction of α_1 and that is the reason for having a large estimate for θ_1. On the other hand, comparing the graphs horizontally does not show any significant difference in the behavior of the response which results in a small estimate for θ_3.

4.1.2.2 Comparison between the SKQ and SK models

In this section, we want to evaluate the effectiveness of sharing the information of the three products in the proposed SKQ model. Therefore, we perform similar fitting process as mentioned in the last section once again, but this time to fit SK models to each product separately, and find the SK estimates of the CT's and their MSE's for the points in $\mathscr{S}_0$. This process is also repeated for 100 macro replications. As a result, using a modeling method (SK or SKQ) for 100 times enables us to find 100 confidence intervals (CI's) for the true response at any evaluation point in $\mathscr{S}_0$ with $\alpha = 0.05$. The coverage probability of the CI's at any point can be estimated as the percentage of the CI's that include the true response. Ideally, we are expecting the percentages be close to 95%. For the sake of graphical and tabular presentations, we set $\alpha_2 = \alpha_3$ and $x = 0.8$, and compare the CI's given by SKQ and SK for the three products in 5 levels of α_1. Table 4.4 presents the coverage probabilities of these 95% CI's over 100 macro replications.

Product	Model	α_1				
		0.1	0.3	0.5	0.7	0.9
Product 1	SK	1.00	1.00	1.00	1.00	1.00
	SKQ	0.92	0.67	0.96	0.97	0.95
Product 2	SK	1.00	1.00	1.00	1.00	1.00
	SKQ	0.94	0.74	0.93	0.96	0.93
Product 3	SK	1.00	1.00	1.00	1.00	1.00
	SKQ	0.95	0.76	0.93	0.96	0.95

TABLE 4.4: The coverage percentage of CI's given SK and SKQ models for the three products

The estimated coverage probabilities of the SK CI's are given in the row marked as SK, and are all equal to 1.00 at the check points, which is much higher than the

expected 95%. However, the estimated coverage probabilities provided by SKQ are presented in the row specified as SKQ, and they are significantly closer to the 95% except for the point that is very close to the bottleneck region. An example of such CI's has been presented in Figure 4.5. This figure clearly shows that the SK models fail to provide tight CI's while the SKQ model provides much narrower CI's. As a result, SKQ models are preferred because the MSE estimates are much smaller than what we obtain in the SK models because of pooling CT information of different products together.

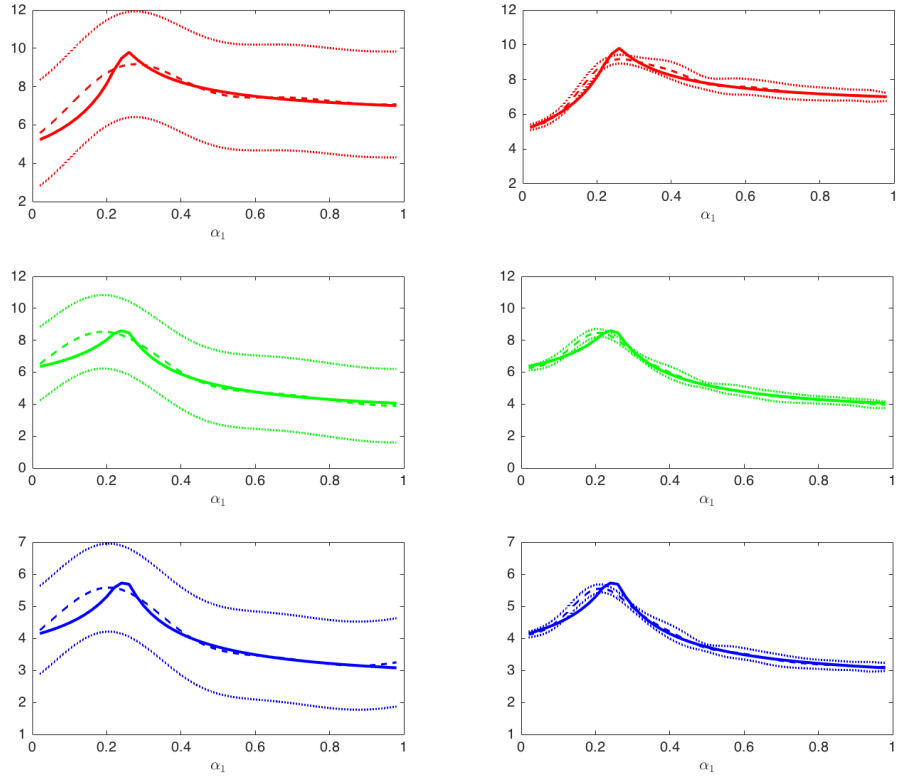


FIGURE 4.5: The 95% CI's provided by SK and SKQ for each product as a function of α_1 in utilization 0.8 and a constant ratio of $\alpha_2 = \alpha_3$

4.2 A Scale-Down Wafer Fab System

In this section, we apply our proposed model to a scale-down semiconductor wafer fab system to estimate its CT-TH-PM surfaces. When performing aggregate production planning, manufacturers usually categorize several possible types of products into a few product families with similar characteristics. This fact along with the benefit of using graphical presentations motivated us to consider a real wafer fab system with three types of product families. An analytical approach similar to what we utilized for the previous study is exploited to perform the capacity/bottleneck analysis of the fab model. This analytical approach also provides an estimate of the system capacity which leads to convert the throughput into system utilization x as mentioned in Section 1.1. Again, we assume that the utilization changes between 0.75 and 0.85 and the PM space is not subject to any further constraint. The latter assumption is not realistic especially for real fab systems, since there are various conditions for meeting the demands in the real world that influence the production mixture. Nevertheless, allowing a wider input space makes the problem of fitting the response more challenging, and gives us a better understanding of the prediction performance of our proposed model. Unfortunately, we can not obtain the true CT in a real wafer fab with the analytical approaches due to the fact that there are many assumptions such as daily demand, batch processing, machine failures, scraps, and so on that can not be characterized with a mathematical approach. Thus, we use the fidelity and flexibility of computer simulation to mimic the behavior of a real wafer fab. More precisely, we run an extensive simulation effort with 500 replications at each point to find the so-called 'nearly true' CT. For the design of experiment, we use the same parameter setting discussed in the previous section, because there is no change in the number of variables. Finally to evaluate the prediction performance of our model, we define a grid of 198 points in the input space of continuous variables as the evaluation set $\mathscr{S}_0$, and find the true responses as discussed earlier. Following the same procedure as in the case of Jackson network, we find 100 SKQ estimates of the CT of each product at any check point, and compute the relative prediction

errors using Equation 4.2. Figure 4.6 shows the box plot of different percentiles of relative prediction errors over the check points using 100 SKQ models.

Among the 198×3 check points, all the relative prediction errors fall within the range of $[-15\%, 11\%]$ with at least 95% of them within $[-8\%, 7\%]$. Therefore, the SKQ model is able to provide a decent prediction of CT of the three products all over the input space of a real waber fab.

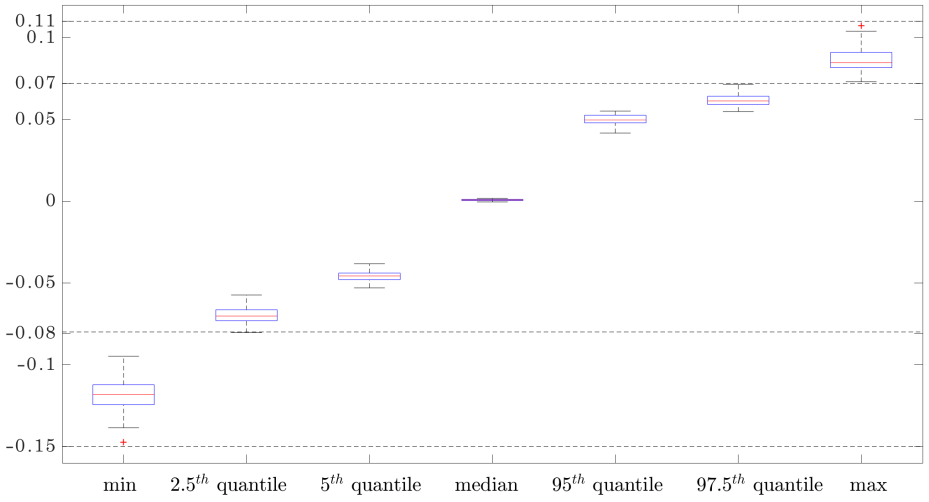


FIGURE 4.6: The box plot of different percentiles of the relative prediction errors in a wafer fab system over 100 macro replications

Chapter 5

CONCLUSIONS

In this study, we developed a metamodeling approach based on the Stochastic Kriging model with Qualitative factors (SKQ) to build CT-TH-PM profiles in semiconductor manufacturing. In such industries, one can use these profiles to answer "what-if" questions in production planning. The conventional models can be categorized into two major classes: analytical approaches which require several simplifications, and simulation-based studies where we perform extensive simulation efforts to obtain CT's for a vast number of scenarios. Adopting a metamodeling approach, we seek to take advantage of the flexibility of computer simulation, and the real-time prediction ability of statistical models.

The advantages of the SKQ-based metamodeling are summarized as follows compared to the existing metamodeling methods in CT-TH-PM quantification. (i) SKQ is able to provide a single model representing the CT-TH-PM response surfaces of all product types. (ii) Unlike many other data mining techniques, SKQ allows for valid statistical inference and hence enables the construction of confidence intervals. (iii) SKQ is able to accommodate heterogeneous variance.

To efficiently estimate CT-TH-PM profiles, a sequential experimental design procedure is developed to carry out simulation experiments in batches. For the initial design, a modified version of a Sequential Maximin algorithm with the conditional Maximin criterion is utilized. This approach provides a space-filling design with

small computational effort. For follow-up designs, we exploit a Particle Swarm Optimization algorithm to maximize the Entropy of the augmented design. A method based on the Sequential Maximin algorithm is proposed to find the initial locations of the particles in the PSO algorithm. Finally, we apply this metamodeling approach to a Jackson network system and a scale-down wafer fab model. It is shown that we obtain high prediction accuracy for most of the points over the entire input space of both examples. Moreover, comparing the CI's provided by SK and SKQ suggest that pooling the information of the qualitative variable (product type) leads us to tighter confidence intervals for the target response.

REFERENCES

Ankenman, B., Nelson, B. L., and Staum, J. (2010). Stochastic kriging for simulation metamodeling. *Operations research*, 58(2):371–382.

Chen, H., Harrison, J. M., Mandelbaum, A., Van Ackere, A., and Wein, L. M. (1988). Empirical evaluation of a queueing network model for semiconductor wafer fabrication. *Operations Research*, 36(2):202–215.

Chen, X., Wang, K., and Yang, F. (2013). Stochastic kriging with qualitative factors. In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 790–801. IEEE Press.

Connors, D. P., Feigin, G. E., and Yao, D. D. (1996). A queueing network model for semiconductor manufacturing. *Semiconductor Manufacturing, IEEE Transactions on*, 9(3):412–427.

Cressie, N. (2015). *Statistics for spatial data*. John Wiley & Sons.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Elsevier.

Fowler, J. W. and Rose, O. (2004). Grand challenges in modeling and simulation of complex manufacturing systems. *Simulation*, 80(9):469–476.

Golchi, S. and Loeppky, J. L. (2015). Space Filing Designs for Constrained Domains. *ArXiv e-prints*.

Henderson, S. G. and Nelson, B. L. (2006). *Handbooks in Operations Research and Management Science: Simulation: Simulation*, volume 13. Elsevier.

Hopp, W. J. (2011). *Supply chain science*. Waveland Press.

Hopp, W. J. and Spearman, M. L. (2001). *Factory Physics: Foundations of Manufacturing Management*. Irwin/McGraw-Hill.

Hopp, W. J., Spearman, M. L., Chayet, S., Donohue, K. L., and Gel, E. S. (2002). Using an optimized queueing network model to support wafer fab design. *IIE Transactions*, 34(2):119–130.

Hung, Y.-F. and Leachman, R. C. (1996). A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *Semiconductor Manufacturing, IEEE Transactions on*, 9(2):257–269.

Jackson, J. R. (1963). Jobshop-like queueing systems. *Management science*, 10(1):131–142.

Jacobs, J. H. et al. (2004). *Performance quantification and simulation optimization of manufacturing flow lines*. Citeseer.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148.

Kelton, W. D. and Law, A. M. (2000). *Simulation modeling and analysis*. McGraw Hill Boston.

Kuehn, P. J. (1979). Approximate analysis of general queuing networks by decomposition. *Communications, IEEE Transactions on*, 27(1):113–126.

Leatherman, E., Dean, A., and Santner, T. (2014). Computer experiment designs via particle swarm optimization. In *Topics in Statistical Simulation*, pages 309–317. Springer.

Loeppky, J. L., Moore, L. M., and Williams, B. J. (2010). Batch sequential designs for computer experiments. *Journal of Statistical Planning and Inference*, 140(6):1452–1464.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4).

Miltenburg, J., Cheng, C. H., and Yan, H. (2002). Analysis of wafer fabrication facilities using four variations of the open queueing network decomposition model. *IIE Transactions*, 34(3):263–272.

Mitchell, T. J. and Morris, M. D. (1992). The spatial correlation function approach to response surface estimation. In *Proceedings of the 24th conference on Winter simulation*, pages 565–571. ACM.

Montgomery, D. C. (2008). *Design and analysis of experiments*. John Wiley & Sons.

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

Qian, P. Z., Wu, H., and Wu, C. J. (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3).

Ramussen, C. and Williams, C. (2006). Gaussian processes for machine learning (adaptive computation and machine learning).

Ranjan, P., Bingham, D., and Michailidis, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4).

Rebonato, R. and Jäckel, P. (2011). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Available at SSRN 1969689*.

Sacks, J. and Schiller, S. (1988). Spatial designs. *Statistical decision theory and related topics IV*, 2(S S):385–399.

Sacks, J., Schiller, S. B., and Welch, W. J. (1989a). Designs for computer experiments. *Technometrics*, 31(1):41–47.

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

Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

Schömig, A. and Fowler, J. (2000). Modeling semiconductor manufacturing operations. In *Proceedings of the 9th ASIM dedicated conference simulation in production and logistics*, pages 55–64.

Shanthikumar, J. G. and Buzacott, J. (1981). Open queueing network models of dynamic job shops. *The International Journal Of Production Research*, 19(3):255–266.

Shanthikumar, J. G., Ding, S., and Zhang, M. T. (2007). Queueing theory for semiconductor manufacturing systems: a survey and open problems. *Automation Science and Engineering, IEEE Transactions on*, 4(4):513–522.

Shewry, M. C. and Wynn, H. P. (1987). Maximum entropy sampling. *Journal of applied statistics*, 14(2):165–170.

Sivakumar, A. I. (1999). Optimization of a cycle time and utilization in semiconductor test manufacturing using simulation based, on-line, near-real-time scheduling system. In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1*, pages 727–735. ACM.

Steinberg, D. M. and Bursztyn, D. (2004). Data analytic tools for understanding random field regression models. *Technometrics*, 46(4):411–420.

Wang, K., Chen, X., Yang, F., Porter, D. W., and Wu, N. (2014). A new stochastic kriging method for modeling multi-source exposure–response data in toxicology studies. *ACS sustainable chemistry & engineering*, 2(7):1581–1591.

Whitt, W. (1983). The queueing network analyzer. *Bell System Technical Journal*, 62(9):2779–2815.

Wu, K., McGinnis, L. F., and Zwart, B. (2007). Compatibility of queueing theory, manufacturing systems and semi standards. In *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pages 501–506. IEEE.

Yang, F. (2010). Neural network metamodeling for cycle time-throughput profiles in manufacturing. *European Journal of Operational Research*, 205(1):172–185.

Yang, F., Ankenman, B., and Nelson, B. L. (2007). Efficient generation of cycle time-throughput curves through simulation and metamodeling. *Naval Research Logistics (NRL)*, 54(1):78–93.

Yang, F., Ankenman, B. E., and Nelson, B. L. (2008). Estimating cycle time percentile curves for manufacturing systems via simulation. *INFORMS Journal on Computing*, 20(4):628–643.

Yang, F., Liu, J., Nelson, B. L., Ankenman, B. E., and Tongarlak, M. (2011). Metamodelling for cycle time-throughput-product mix surfaces using progressive model fitting. *Production Planning and Control*, 22(1):50–68.

Zhou, Q., Qian, P. Z., and Zhou, S. (2011). A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics*, 53(3).