Holistic modeling and analysis of multistage manufacturing processes with sparse effective inputs and mixed profile outputs

Andi Wang & Jianjun Shi

To cite this article: Andi Wang & Jianjun Shi (2020): Holistic modeling and analysis of multistage manufacturing processes with sparse effective inputs and mixed profile outputs, IISE Transactions, DOI: 10.1080/24725854.2020.1786197

To link to this article: https://doi.org/10.1080/24725854.2020.1786197

IISE TRANSACTIONS
https://doi.org/10.1080/24725854.2020.1786197

Holistic modeling and analysis of multistage manufacturing processes with sparse effective inputs and mixed profile outputs

Andi Wang and Jianjun Shi

Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT
In a Multistage Manufacturing Process (MMP), multiple types of sensors are deployed to collect intermediate product quality measurements after each stage of manufacturing. This study aims at modeling the relationship between these quality outputs of mixed profiles and sparse effective process inputs. We propose an analytical framework based on four process characteristics: (i) every input only affects the outputs of the same and the later stages; (ii) the outputs from all stages are smooth functional curves or images; (iii) only a small number of inputs influence the outputs; and (iv) the inputs cause a few variation patterns on the outputs. We formulate an optimization problem that simultaneously estimates the effects of process inputs on the outputs across the entire MMP. An ADMM consensus algorithm is developed to solve this problem. This algorithm is highly parallelizable and can handle a large amount of data of mixed types obtained from multiple stages. The ability of this algorithm in estimations, selecting effective inputs, and identifying the variation patterns of each stage is validated with simulation experiments.

ARTICLE HISTORY
Received 27 February 2020
Accepted 14 June 2020

KEYWORDS
Multistage manufacturing process; profile data; sparse data

1. Introduction

Contemporary Multistage Manufacturing Processes (MMPs) are usually equipped with advanced sensing systems that collect both a large number of process input variables and intermediate product quality measurements from each stage of an MMP. Examples of inputs include the process parameters set by the engineers, the environmental variables, and the external events that occurred to the process. The variation of the process output, the intermediate product quality measurements in every stage, is potentially caused by the variation of certain inputs of the processes. This article performs a root cause analysis of the variability in the outputs of MMPs by associating them with specific process inputs in all stages. Specifically, we aim at answering three interrelated questions: (i) which effective inputs relate to the variations of the outputs? (ii) what are the variation patterns of the outputs caused by these inputs? (iii) how each individual process input affects the manufacturing process? Answering these questions leads to a better understanding of the process variabilities.

The statistical analysis of MMPs has been conducted for decades (Shi, 2006; Li and Shi, 2007; Jin and Shi, 2012). However, there are two major limitations of the existing analytical methods. First, they are unable to be applied to intermediate product quality measurements of mixed types of data in an MMP, which are increasingly common in data-rich manufacturing environments. Here, "mixed types of data" means that the data collected from different stages have different dimensions and distinct characteristics. As an example, a semiconductor manufacturing process consists of hundreds of stages, including deposition, lithography, plasma etching, ion-implantations, chemical-mechanical polishing, etc. (Nishi and Doering, 2000). In different stages, the corresponding outputs can be of image types (e.g., spatial data such as film thickness of each layer at multiple locations on the wafer (De Witte *et al.*, 2003), multivariate random fields such as the alignment error at hundreds of positions on a wafer's surface (Huang *et al.*, 2008), and images of the etched trenches captured by a scanning electron microscope (Lee *et al.*, 2006)), and/or assorted functional curves (e.g., the temperature, pressure, and radio-frequency curves during the reaction processes). As will be discussed in the next section, Stream of Variation (SOV) modeling approaches (Shi, 2006) based on a state-space model is generally not suitable for an MMP with mixed type of data. There is a lack of appropriate analytical methods for MMPs where output sensing data contains mixed types of data, such as images or functional curves.

Second, existing analytical methods for MMPs cannot handle a large number of inputs associated with each stage of an MMP. In the example of semiconductor manufacturing, tens of control variables adjust the exposure system in a lithography step. For one thing, a large number of inputs calls for efficient and parallel implementation of the model estimation algorithm. For another, we need to identify the inputs that are truly related to the outputs and establish the connections between them.

CONTACT Andi Wang andi.wang@gatech.edu
This article has been corrected with minor changes. These changes do not impact the academic content of the article.

In this article, we propose a system-level modeling framework to solve the diagnostic problem for MMPs that generate intermediate product quality measurements of mixed types and different dimensions and sparse effective inputs. Based on the characteristics of an MMP, we propose the following assumptions that facilitate our modeling approach:

1. **Cascading assumption**: An input in one stage only affects the outputs generated from that stage and the downstream stages. The cascading assumption is rooted in the directional error propagation among stages: the input variation from one stage not only affects the quality measurement of the current stage, but its effect can propagate to the next stage, and further downstream stages. However, the input from one downstream stage cannot affect the quality measurement of its upstream stage.
2. **Mixed data types:** The outputs generated from different stages of a process may be collected through different metrology systems, have different dimensions, and thus have different characteristics. We illustrate our modeling approach by assuming that each stage generates one of two types of outputs: smooth functional curves and smooth images. Our idea has the potential to be applied to measurements with different structural assumptions (see the discussion in Section 3.4).
3. **Sparsity assumption**: The potential root causes of the variability of the output is driven by a small number of *effective inputs*. We assume that the effective inputs are sparse: they compose a very small portion of all inputs from an MMP.
4. **Low-rank assumption:** In reality, the measurements from each stage of an MMS may have multiple sources of variations. Within a short period of time, the potential root causes in an MMP is limited, and thus only a small number of dominant variation patterns significantly impact quality and system performance.

Leveraging these assumptions, we propose a *holistic* modeling framework for MMPs. The word "holistic" means that the model describes the entire manufacturing system composed of all stages, and we estimate the process parameters that represent the relationships between all process inputs and outputs simultaneously. An optimization problem is formulated for the estimation process, and its objective function contains the magnitude of the predictive error of each stage, the smoothness of the functional curves or image outputs, the sparsity of the effective inputs, and the number of variability patterns caused by the inputs. From the estimation procedure, we can solve three diagnostic problems for MMPs: identify the effective inputs, identify the variation patterns of the outputs, and describe how each input affects the output of each stage. In Section 4, an illustrative example is provided via simulation studies on how the proposed method effectively solves the above three problems.

To our knowledge, this study is the first one that proposes a holistic modeling and analysis framework for an MMP that generates assorted types of data. It simultaneously answers three questions involving the effective inputs, the variation patterns in the outputs, and their connections. The idea behind the proposed method is extendable to a wide range of MMPs that involve: (i) a comprehensive set of inputs, and (ii) process outputs of mixed types of data with limited variation patterns from each stage. Also, our model estimation method based on an Alternating Direction Method of Multipliers (ADMM) is highly parallelizable, and thus guarantees high computational efficiency for an MMP with more stages.

The remaining parts of this article are organized as follows. In Section 2, we review related literature to highlight the necessity of this research. In Section 3, we present the mathematical description of our problem, formulate the optimization problem, and propose the algorithm for solving this problem. In Section 4, the methods proposed in Section 3 are validated through simulation experiments. Section 5 concludes the article.

2. Literature review

The modeling and statistical analysis of an MMP have been investigated for decades. Since the mid-1990s, state-space models have been proposed to describe the SOV in MMPs for assembly and machining processes (Shi, 2006). Based on the state-space model, estimation-based diagnostics methods have been proposed to find the connection between the product quality measurements from each stage and the sources of errors (Apley and Shi, 1998). In most of the existing literature, the product quality measurements are 3D coordinates of a set of critical points on a fabricated part. Based on the engineering design and physics principles (Ding *et al.*, 2000), the proposed state-space model can accurately describe the error propagation between stages in an MMP. For modeling complex MMPs with mixed profile outputs, two issues make the state-space modeling approach infeasible. First, the state-space model describes the status of the manufacturing process by using a state vector. The state vector cannot be defined when the product quality measures are represented by functional curves or images. Second, the state-space model assumes that the state at stage k is solely determined by the state at stage $k-1$, and not related to previous stages $k-2, k-3, \dots$ However, in MMPs that resemble semiconductor manufacturing processes, the output from one stage may relate to the inputs from more than one previous stage. Therefore, we do not adopt the state-space modeling approach, but instead propose a regressive approach that directly represents the relationship between the output of each stage and the inputs from all previous stages.

In order to model the MMPs that generate profile data (e.g., functional curves or images) in each stage, we extend the literature on modeling the relationships between profile inputs and outputs. Regression between smooth profile data and scalars is a part of functional data analysis (Ramsay, 2005), and studies such as Li *et al.* (2020) tailor this technique for engineering applications. Recently, some studies (Yan *et al.*, 2014; Gahrooei *et al.*, 2020; Yue *et al.*, 2020) use tensors to describe the profile of the same size and use tensor regression techniques to model the relationship between profile inputs and outputs. Many studies that apply functional and tensor regression penalize the parametric vectors and matrices for promoting certain characteristics of the input and output profiles, including sparsity (Tibshirani, 1996), continuity and smoothness (Ramsay, 1988), low-rank (Yuan *et al.*, 2007), and the flatness among neighboring elements (Tibshirani *et al.*, 2005). They are used collectively for anomaly detection (Yan *et al.*, 2017), multiple change point detections based on sparse signal dependency (Zhang *et al.*, 2018a), and so forth, to improve the estimation accuracy and identify the effective inputs and the variation patterns caused by them. This article also applies penalizations to represents the characteristics of the mixed profile discussed in Section 1.

To solve the diagnostic problems of the MMPs that generate mixed profile outputs, we need to integrate the above profile data modeling techniques into a model that specifies how the inputs from one stage propagate to the follow-up stages. There is no well-established theory for modeling and analysis of MMPs with mixed profile outputs and a large number of inputs that answers all three questions in the Introduction, given the complex input–output interactions. Specifically, if we use existing profile data analysis approaches to build separate models for the relationship between the output of every stage k and the inputs from stage $1, \ldots, k$, these models may result in different sets of effective inputs, and thus may result in different effective inputs. Another common practice of analyzing these MMPs nowadays is to adopt a two-step procedure: we first extract a set of features from the process output profiles in every stage and then perform the analysis of the process based on these selected features. For example, Zhang *et al.* (2018b) followed this procedure to perform anomaly detection from data from a single stage. However, the second step is usually sensitive to the set of features selected.

Our model estimation process involves solving an optimization problem with multiple penalization terms. We use an ADMM consensus algorithm for this purpose. Its general framework is introduced in Parikh and Boyd (2014) and Boyd *et al.* (2011). We cast our problem into an appropriate form and adopt the ADMM consensus method to solve it.

3. Holistic modeling and analysis framework for an MMP generating profiles and images

In this section, we first describe the data generated from an MMP, then propose a holistic modeling and analysis framework for the MMP. We present how to solve the modeling and estimation problem using an ADMM consensus algorithm. We also discuss the selection of the tuning parameters and the possible variation of the problem formulation.

3.1. Data scenario and problem description

We assume that the process outputs from each stage either includes multiple functional curves of the same length or an image. The process output of stage k can be written as $\mathbf{Y}_k \in \mathbb{R}^{m_k \times n_k}$. If the output from stage k is an image, $\mathbf{Y}_k$ represents an image of size $m_k \times n_k$. If stage k generates functional curves, $\mathbf{Y}_k$ represents m_k curves of length n_k. The set $\mathcal{I}$ includes the indices of stages that generate image data, and $\mathcal{S} = \{1, \ldots, K\} - \mathcal{I}$ represent the stages that generate functional curves. All outputs for product n are thus described by $\mathcal{Y}^{\{n\}} = \left(\mathbf{Y}_1^{\{n\}}, \ldots, \mathbf{Y}_K^{\{n\}}\right)$. Throughout this article, we use curly brackets $\{\cdot\}$ to identify the product number.

Note that the structure of $\mathcal{Y}^{\{n\}}$ describes the general data structure for the intermediate product quality data generated from an MMP of multiple data types and their dimensions. This structure is similar to a C struct or MATLAB® cell: the data generated from all stages are of different dimensions. If the size of matrices $\mathbf{Y}_1, \ldots, \mathbf{Y}_K$ are the same, $\mathcal{Y}^{\{n\}}$ can be seen as tensor data (Yue *et al.*, 2020). We assume that the data generated from all stages have different structures, in the sense that they may represent either images or functional curves, so that special considerations in data analytics are required. Finally, we note that in other applications, the process outputs can be even more complicated. For example, the data from each stage may include multiple images of different sizes, groups of functional curves of different sizes, or other structured data types such as spatial measurements or point clouds. It will become clear in Section 3.4 that the methodology proposed in this article can potentially be extended to such scenarios.

We further assume that there are q_k inputs from stage k that may affect the process, represented by $\mathbf{u}_k = (u_{k1}, \ldots, u_{kq_k})$, $k = 1, \ldots, K$. For simplicity, we assume that the effect of the inputs on the outputs is always linear. The treatment of nonlinear effects is briefly discussed in Section 3.4.2. Based on a linear model, the effect of the process inputs on the process outputs can be described by equation (1):

$$\mathbf{Y}_k = \mathbf{B}_{k0} + \sum_{i=1}^{k} \sum_{j=1}^{q_i} u_{ij} \mathbf{B}_{ij,k} + \mathbf{E}_k. \tag{1}$$

In model (1), the parametric matrix $\mathbf{B}_{ij,k}$ is of size $\mathbb{R}^{m_k \times n_k}$. It is referred to as an *effect matrix*, as it describes the effect of the input u_{ij} on the process outputs measured from stage k. The collection of effect matrices is denoted as $\mathcal{B} = \left\{\mathbf{B}_{ij,k} : 1 \le i \le \mathrm{k} \le K, 1 \le j \le q_i\right\}$. The matrices $\mathbf{B}_{k0}$ are called *offset matrices*, and the set of all offset matrices is denoted by $\mathcal{B}_0 = \{\mathbf{B}_{k0} : 1 \le k \le K\}$. Here $\mathbf{E}_k$ is a matrix representing the modeling error of the stage k, and we assume that every entry of $\mathbf{E}_k$ is with mean 0, and variance $\sigma^2_{E,k}$. The cascading effect of the process inputs is inherently reflected from this model, as the input variable u_{ij} from stage i affects the output from stage k, $\mathbf{Y}_k$ only if $i \le k$. Other three assumptions discussed above can be cast into specifications on the model parameter $\mathcal{B}_0$ and $\mathcal{B}$:

1. As $\mathbf{B}_{ij,k}$ and $\mathbf{B}_{k0}$ represent the effect of parameter u_{ij} on such process outputs, they share the same characteristic as the curve or image data in stage k. Specifically, if the process outputs from stage k are multiple smooth

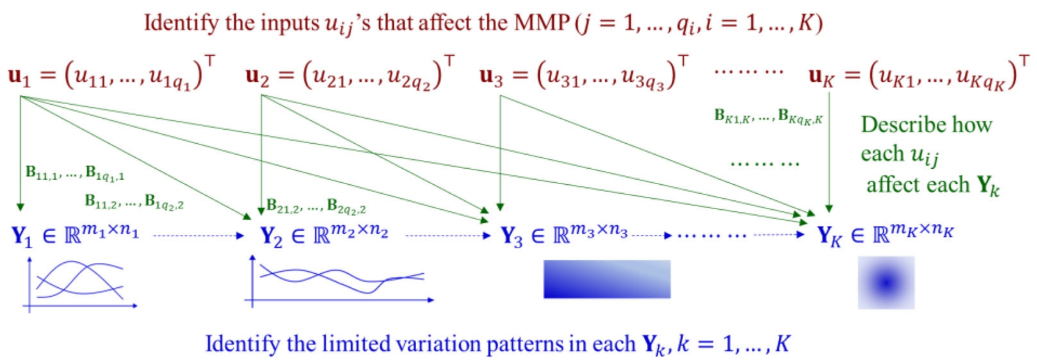
Identify the inputs u_{ij}'s that affect the MMP ($j = 1,\dots,q_i, i = 1,\dots,K$)
$\mathbf{u}_1 = (u_{11},\dots,u_{1q_1})^\top$
$\mathbf{u}_2 = (u_{21},\dots,u_{2q_2})^\top$
$\mathbf{u}_3 = (u_{31},\dots,u_{3q_3})^\top$
$\mathbf{u}_K = (u_{K1},\dots,u_{Kq_K})^\top$
$\mathbf{B}_{11,1},\dots,\mathbf{B}_{1q_1,1}$
$\mathbf{B}_{11,2},\dots,\mathbf{B}_{1q_2,2}$
$\mathbf{B}_{21,2},\dots,\mathbf{B}_{2q_2,2}$
$\mathbf{B}_{K1,K},\dots,\mathbf{B}_{Kq_K,K}$
Describe how each u_{ij} affect each $\mathbf{Y}_k$
$\mathbf{Y}_1 \in \mathbb{R}^{m_1\times n_1}$
$\mathbf{Y}_2 \in \mathbb{R}^{m_2\times n_2}$
$\mathbf{Y}_3 \in \mathbb{R}^{m_3\times n_3}$
$\mathbf{Y}_K \in \mathbb{R}^{m_K\times n_K}$
Identify the limited variation patterns in each $\mathbf{Y}_k, k = 1,\dots,K$

Figure 1. The illustration of the model (1).

curves, every row in $\mathbf{B}_{ij,k}$ and $\mathbf{B}_{k0}$ corresponds to a curve, and thus two elements whose indices are close in each row should have similar values. If the stage k generates smooth images, any two elements whose indices are close in $\mathbf{B}_{ij,k}$ or $\mathbf{B}_{k0}$ should have similar values.

2. The sparsity assumption indicates that most u_{ij} ($1 \le i \le K$ and $1 \le j \le q_i$) are associated with effect matrices $\mathbf{B}_{ij,k} = \mathbf{O}$ for all $k = 1,\dots,K$.
3. The low-rank assumption indicates that the variation patterns caused by all effective u_{ij} lie in a low-dimensional subspace. Therefore, the matrix $\mathbf{B}_{\cdot\cdot,k}$ is of low rank, where

$$\mathbf{B}_{\cdot\cdot,k} = \left[\mathrm{vec}(\mathbf{B}_{11,k}), \mathrm{vec}(\mathbf{B}_{12,k}),\dots,\mathrm{vec}(\mathbf{B}_{1q_1,k}),\dots,\mathrm{vec}(\mathbf{B}_{kq_k,k})\right] \in \mathbb{R}^{(m_k n_k)\times\left(\sum_{i=1}^{k} q_i\right)}.$$

Here, $\mathbf{B}_{\cdot\cdot,k}$ constitutes the effects of $\sum_{i=1}^{k} q_i$ inputs from the first k manufacturing stages. The vectorization operator $\mathrm{vec}(\cdot)$ transforms the $m_k \times n_k$ matrix to vectors of size $m_k n_k \times 1$.

3.1.1. Interpretation of the model and its use for root cause diagnostics

The model (1) we proposed above can be illustrated by using Figure 1. In this figure, we can see that every input u_{ij} from stage i only affects the output in stage i and the following stages $i+1,\dots,K$, indicated by the effect matrices $\mathbf{B}_{ij,k}$ and the arrows that point from an input to an output. Although we do not explicitly model how the output from stage k affects the output of stage $k+1$ as in the SOV modeling approach (shown as dashed arrows in Figure 1), we acknowledge that the error propagation between the output stages exists, and embed this consideration into the cascading assumption: The effective input from stage i may not only affect stage i itself, but it may further influence the outputs of downstream stages $i+1, i+2,$ etc. The proposed regressive approach has several benefits. First, it enables existing multilinear regression and functional regression techniques applied to the MMP. Second, it gives an explicit description of how the input from one stage influences the outputs of downstream stages. Third, it can be easily extended for other types of process outputs with different characteristics, as specified in Section 3.4.2.

We can use the proposed model (1) to solve the diagnostic problem of the MMP. After we obtain the inputs $\mathbf{u}_1^{\{n\}},\dots,\mathbf{u}_K^{\{n\}}$ and the outputs $\mathcal{Y}^{\{n\}}$ for samples $n = 1,\dots,N$, we can estimate the effect matrices in $\mathcal{B}$ in the model (1) such that the cascading assumption, mixed data type assumption, sparsity assumption, and low-rank assumption are satisfied. After the estimation is obtained, three questions of the diagnostics can be readily answered. Specifically, we can:

1. Identify the effective inputs (e.g., root causes): the input u_{ij} is identified as an effective input if $\mathbf{B}_{ij,k} \ne \mathbf{O}$ for some $k \in \{i, i+1,\dots,K\}$.
2. Identify the output variation patterns: the variation driven by all inputs in stage $1,\dots,k$ for the output of stage k is described by the linear subspace spanned by all $\mathbf{B}_{ij,k}: 1 \le i \le k, 1 \le j \le q_i$, whose dimension is given as the rank of $\mathbf{B}_{\cdot\cdot,k}$
3. Determine the effect of a specific inputs on the outputs: how each input u_{ij} affects the output of stage k is described by $\mathbf{B}_{ij,k}$.

From the model, we can see that there are many parameters to be estimated in the modeling efforts. However, the overfitting problem can be avoided by the penalties applied to these parameters based on our model assumptions, as detailed in the next section. If we know that the possible causal structures between input variables and quality measurements in advance from domain knowledge, we may further limit the number of parameters to be estimated. For example, if we know that the inputs u_{ij} may only affect the output from stage i, $i+1,\dots,k_{ij}$, we can add another constraint $\mathbf{B}_{ij,k} = \mathbf{O}$ for all $k > k_{ij}$.

In the following sections, we describe how to formulate an optimization problem to solve the parameters in $\mathcal{B}_0$ and $\mathcal{B}$, and how to solve them numerically.

3.2. Problem formulation

The objective of this study is to describe how the inputs $u_{11}, \dots, u_{Kq_K}$ affect the outputs $\left(\mathbf{Y}_1^{\{n\}}, \dots, \mathbf{Y}_K^{\{n\}}\right)$ in all stages. It is achieved by obtaining an estimation of $\mathcal{B}$ and $\mathcal{B}_0$ with the characteristics described in the previous subsection. To obtain the estimation, we solve an optimization problem that minimizes the sum of the prediction error of the process outputs and the penalties specified by each assumption. These terms are detailed as follows.

Prediction error of the process outputs. From the model (1), the prediction error for the process outputs of product n from stage k can be represented as

$$\mathbf{E}_k^{\{n\}} = \mathbf{Y}_k^{\{n\}} - \mathbf{B}_{k0} - \sum_{i=1}^{k}\sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k},$$

and the prediction accuracy can be represented as $\|\mathbf{E}_k^{\{n\}}\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. The prediction accuracy across all K stages is then represented as

$$\mathcal{L}(\mathcal{B}, \mathcal{B}_0) = \sum\nolimits_{n=1}^{N} \sum\nolimits_{k=1}^{K} \|\mathbf{E}_k^{\{n\}}\|_F^2 = \sum_{n=1}^{N}\sum_{k=1}^{K} \left\| \mathbf{Y}_k^{\{n\}} - \mathbf{B}_{k0} - \sum_{i=1}^{k}\sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k} \right\|_F^2. \tag{2}$$

In this expression, we sum up the loss $\|\mathbf{E}_k^{\{n\}}\|_F^2$ corresponding to all stages $1, \dots, K$, to determine the effective inputs and estimate their effects using the information from all stages. Note that every effective input affects the outputs in all later stages. Thus, we need to incorporate the information from all stages to identify them. Taking different magnitudes of the error and different numbers of elements of all stages into consideration, one may incorporate a weight $1/\left(\hat{\sigma}_{E,k}^2 m_k n_k\right)$ to the term corresponding to stage k, where the parameter $\hat{\sigma}_{E,k}^2$ is a rough estimation obtained through smoothing the outputs of a subset of samples from every stage k and calculate the mean-squared error. Under this setting, the formulation is only slightly modified, and the solution framework remains the same.

The effects of each input are smooth. Assume that stage $k \in \mathcal{S}$ generates functional curves. According to Section 3.1, the elements on every row of $\mathbf{B}_{ij,k}$ should form a smooth function. To enhance the smooth property of the curves and thus increase the estimation accuracy, we propose the following penalization for these stages, similar to the smooth component in Yan *et al.* (2017):

$$p_1(\mathcal{B}, \mathcal{B}_0) = \sum_{k \in \mathcal{S}} \lambda_{1,k} \left[\sum_{m=0}^{m_k} \|\mathbf{D}_S \mathbf{B}_{k0}(m,:)\|_2^2 + \sum_{i=1}^{k}\sum_{j=1}^{q_i}\sum_{m=1}^{m_k} \|\mathbf{D}_S \mathbf{B}_{ij,k}(m,:)\|_2^2 \right] \tag{3}$$

In this term, $\mathbf{B}_{ij,k}(m,:)$ describes the effect of u_{ij} on the mth functional curve generated from stage k. $\mathbf{D}_S$ is a modified 1D second-order difference matrix for smoothing curves, with modified Neumann boundary condition (O'Sullivan, 1991):

$$\mathbf{D}_S = \begin{bmatrix} -1 & 1 & & & \mathbf{0} \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ \mathbf{0} & & & 1 & -1 \end{bmatrix}.$$

The expression $\mathbf{D}_S \mathbf{B}_{ij,k}(m,:)$ gives a discretized approximation of function norm $\|f''(x)\|_2^2$ (Ramsay, 1988), where $f(x) = \mathbf{B}_{ij,k}(m,x)$. As we will see in Algorithm 2 and Algorithm 3, the choice of the boundary condition (the first and the latest row of $\mathbf{D}_S$) enables efficient computation. Here $\lambda_{1,k}$ is selected to control the degree of smoothing for the signals in stage k. Motivated by thin-plate splines (Wahba, 1990), similar penalization term is defined for the stages that generate a smooth image:

$$p_2(\mathcal{B}, \mathcal{B}_0) = \sum_{k \in \mathcal{I}} \lambda_{2,k} \left[\operatorname{vec}(\mathbf{B}_{k0})^\top \mathbf{R}_I \operatorname{vec}(\mathbf{B}_{k0}) + \sum_{i=1}^{k}\sum_{j=1}^{q_i} \operatorname{vec}(\mathbf{B}_{ij,k})^\top \mathbf{R}_I \operatorname{vec}(\mathbf{B}_{ij,k}) \right]. \tag{4}$$

Here $\operatorname{vec}(\cdot)$ transforms the image to a $m_i \times n_i$ vector, and $\mathbf{R}_I$ is a discretized version of the operator:

$$\mathcal{R}(g) = \int_{\mathbb{R}^2} \left[\left(\frac{\partial^2 g}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 g}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 g}{\partial y^2}\right)^2 \right] dxdy,$$

that defines the "roughness" of bivariate function g (Buckley, 1994). The closed-form expression of the roughness matrix $\mathbf{R}_I$ for an $m \times n$ image is derived and represented in Section 3 of Buckley (1994):

$$\mathbf{R}_I = \left(\mathbf{C}_m^\top \otimes \mathbf{C}_n^\top\right)\left(\mathbf{M}_m^2 \otimes \mathbf{I}_n + 2\mathbf{M}_m \otimes \mathbf{M}_n + \mathbf{I}_m \otimes \mathbf{M}_n^2\right)\left(\mathbf{C}_m \otimes \mathbf{C}_n\right),$$

where $\mathbf{C}_n$ is discrete cosine transform of order n, $\mathbf{I}_n$ is an identity matrix of order n, $\mathbf{M}_n$ is a diagonal matrix whose diagonal elements are $\mu_{i,n} = 2[1 - \cos\{\pi(i-1)/n\}]$, $i = 1, \dots, n$, and "$\otimes$" represents the Kronecker product. However, we will see later that the image $\mathbf{R}_I$ does not need to be constructed explicitly.

The effects of the inputs are sparse. In model (1), "$\mathbf{B}_{ij,k} = \mathbf{O}$ for all k" is satisfied for most (i,j) pairs with $1 \le i \le k$ and $1 \le j \le q_i$. Motivated by the Group lasso algorithm (Yuan and Lin, 2006), an ℓ_2 penalty is applied to all elements in $\mathcal{B}$ that involve the input u_{ij}. Specifically, for u_{ij} we define a long vector:

$$\mathbf{B}_{ij,\cdot} = \left(\frac{1}{\sqrt{C_i}} \operatorname{vec}(\mathbf{B}_{ij,i}); \frac{1}{\sqrt{C_{i+1}}} \operatorname{vec}(\mathbf{B}_{ij,i+1}); \dots; \frac{1}{\sqrt{C_k}} \operatorname{vec}(\mathbf{B}_{ij,k}) \right),$$

that constitutes all elements in $\mathcal{B}$, characterizing how u_{ij} affects the outputs. Here, the parameters $C_i = m_i n_i, \dots, C_k = m_k n_k$ are the number of elements in $\mathbf{B}_{ij,i}, \mathbf{B}_{ij,i+1}, \dots, \mathbf{B}_{ij,k}$ that adjust the weights of the components to make the effects of each stage have comparable norms. The penalization term is then defined based on the $\|\mathbf{B}_{ij,\cdot}\|_2$, given as

$$p_3(\mathcal{B}) = \sum_{i=1}^{K} \left[\lambda_{3,i} \sum_{j=1}^{q_i} \|\mathbf{B}_{ij,\cdot}\|_2 \right]. \tag{5}$$

Here $\lambda_{3,i}$ controls the level of the sparsity of effective inputs from stage i. With more effective inputs from stage i, $\lambda_{3,i}$ should be selected smaller.

Variation caused by inputs is of low rank. As presented in Section 3.1, the matrix $\mathbf{B}_{\cdot\cdot,k}$ should be of low rank. A heuristic for solving rank minimization problems is by minimizing the nuclear norm of a matrix (Fazel *et al.*, 2001), and the nuclear norm penalization has been proposed for reduced-rank regression (Yuan *et al.*, 2007). We borrow this idea and apply the following penalization term to limit the number of variation patterns of each stage, resulted from all inputs that affect it:

$$p_4(\mathcal{B}) = \sum_{k=1}^{K} \lambda_{4,k} \|\mathbf{B}_{\cdot\cdot,k}\|_*. \tag{6}$$

The overall objective function is given as the sum of the prediction error of the process outputs and four regularization terms $p_1(\mathcal{B})$, $p_2(\mathcal{B})$, $p_3(\mathcal{B})$ and $p_4(\mathcal{B})$ listed in (3)–(6). Therefore, our objective is to solve the following optimization problem:

$$\underset{\mathcal{B},\mathcal{B}_0}{\text{minimize}} \quad \mathcal{L}(\mathcal{B},\mathcal{B}_0) + p_1(\mathcal{B},\mathcal{B}_0) + p_2(\mathcal{B},\mathcal{B}_0) + p_3(\mathcal{B}) + p_4(\mathcal{B}). \tag{7}$$

3.3. Problem solution

Note that formulation (7) is a convex problem, lower bounded by zero. Therefore, it has an optimal solution. This problem has two characteristics. First, the problem has many decision variables, and thus a highly parallel algorithm is desired. Second, its objective function contains multiple non-differentiable additive components. For this reason, we apply an ADMM consensus algorithm to solve this problem (Parikh and Boyd, 2014).

To cast the formulation (7) into the ADMM consensus framework, we introduce four copies of parameter $\mathcal{B}$, namely $\mathcal{B}^{(1)}, \mathcal{B}^{(2)}, \mathcal{B}^{(3)}$ and $\mathcal{B}^{(4)}$, and two copies of parameters $\mathcal{B}_0$: $\mathcal{B}_0^{(1)}$ and $\mathcal{B}_0^{(2)}$. Then the formulation (7) is equivalent to the formulation (8) below:

$$\min f(\tilde{\mathcal{B}}) + g(\tilde{\mathcal{B}}). \tag{8}$$

In formulation (8) $\tilde{\mathcal{B}} = \left(\mathcal{B}_0^{(1)}, \mathcal{B}_0^{(2)}, \mathcal{B}^{(1)}, \mathcal{B}^{(2)}, \mathcal{B}^{(3)}, \mathcal{B}^{(4)}\right)$ represents the collection of augmented parameters, the function $f(\tilde{\mathcal{B}}) = \mathcal{L}\left(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}\right) + p_1\left(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}\right) + p_2\left(\mathcal{B}_0^{(2)}, \ \mathcal{B}^{(2)}\right) + p_3\left(\mathcal{B}^{(3)}\right) + p_4\left(\mathcal{B}^{(4)}\right)$. The function $g(\tilde{\mathcal{B}}) = I_{\mathcal{B}^{(1)}=\mathcal{B}^{(2)}=\mathcal{B}^{(3)}=\mathcal{B}^{(4)}}(\tilde{\mathcal{B}}) \cdot I_{\mathcal{B}_0^{(1)}=\mathcal{B}_0^{(2)}}(\tilde{\mathcal{B}})$ specifies that the copies are of the same values, where

$$I_A(x) = \begin{cases} 0 & x \in A \\ +\infty \text{ if } & x \notin A \end{cases}$$

is the indicator function. The formulation (8) is solved with a general framework of ADMM listed in Algorithm 1.

Algorithm 1. General framework of ADMM

```
Initialize $\tilde{\mathcal{Z}}$ and $\tilde{\mathcal{U}}$ as structs with the same shape as $\tilde{\mathcal{B}}$. Set all elements to 0.
Do:
    Set $\tilde{\mathcal{B}}_{\text{prev}} \leftarrow \tilde{\mathcal{B}}$, $\tilde{\mathcal{Z}}_{\text{prev}} \leftarrow \tilde{\mathcal{Z}}$ and $\tilde{\mathcal{U}}_{\text{prev}} \leftarrow \tilde{\mathcal{U}}$
    $\tilde{\mathcal{B}} \leftarrow \text{prox}_{\eta f}[\tilde{\mathcal{Z}} - \tilde{\mathcal{U}}]$ (Step 1)
    $\tilde{\mathcal{Z}} \leftarrow \text{prox}_{\eta g}[\tilde{\mathcal{B}} + \tilde{\mathcal{U}}]$ (Step 2)
    $\tilde{\mathcal{U}} \leftarrow \tilde{\mathcal{U}} + \tilde{\mathcal{B}} - \tilde{\mathcal{Z}}$ (Step 3)
Until $\|\tilde{\mathcal{U}} - \tilde{\mathcal{U}}_{\text{prev}}\| < \epsilon$ and $\|\tilde{\mathcal{Z}} - \tilde{\mathcal{Z}}_{\text{prev}}\| < \epsilon$.
```

In this algorithm, the summation and subtraction of two structs are naturally defined as adding and subtracting each corresponding element. The parameter η specifies the step size, and $\text{prox}_h(\mathbf{x})$ is the proximal operator, defined as

$$\text{prox}_h(\mathbf{x}) = \underset{\mathbf{y}}{\text{argmin}} \left\{ h(\mathbf{y}) + \frac{1}{2}\|\text{vec}\,(\mathbf{x} - \mathbf{y})\|_2^2 \right\},$$

where the operator vec $(\cdot)$ in the second term transforms the struct to a long vector.

To perform this optimization algorithm, we evaluate the proximal operator in **Step 1** and **Step 2** in the following two subsections.

3.3.1. Evaluating the proximal operator in Step 1

The function $f(\tilde{\mathcal{B}})$ is the summation of four components, and the elements involved in every component do not overlap. The separable property of the proximal operator (Parikh and Boyd, 2014) states that:

$$\text{prox}_h(\mathbf{x}_1, \mathbf{x}_2) = \left(\text{prox}_{h_1}(\mathbf{x}_1), \text{prox}_{h_2}(\mathbf{x}_2)\right), \tag{9}$$

if $h(\mathbf{x}_1, \mathbf{x}_2) = h_1(\mathbf{x}_1) + h_2(\mathbf{x}_2)$. Therefore, the proximal operator of $\text{prox}_{\eta f}[\tilde{\mathcal{B}}]$ is determined by that of the each components, evaluated as follows.

The proximal operator of $\mathcal{L}\left(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}\right)$. The first term

$$\mathcal{L}\left(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} \|\mathbf{Y}_k^{\{n\}} - \mathbf{B}_{k0}^{(1)} - \sum_{i=1}^{k}\sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k}^{(1)}\|_F^2$$
$$= \sum_{k=1}^{K}\sum_{v=1}^{m_k}\sum_{w=1}^{n_k} S_{k,u,v}\left(\mathcal{B}_{k,v,w}^{(1)}\right),$$

where

$$S_{k,v,w}(\mathcal{B}_{k,v,w}^{(1)}) = \sum_{n=1}^{N}\left(\mathbf{Y}_k^{\{n\}}(v,w) - \mathbf{B}_{k0}^{(1)}(v,w) - \sum_{i=1}^{k}\sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k}^{(1)}(v,w)\right)^2, \tag{10}$$

is the summation of least square components that involve disjoint sets of elements

$$\mathcal{B}_{k,v,w}^{(1)} = \left\{\mathbf{B}_{k0}^{(1)}(v,w)\right\} \cup \left\{\mathbf{B}_{ij,k}^{(1)}(v,w) : i = 1,\ldots,k; j = 1,\ldots,q_i\right\},$$

$k = 1, \ \ldots, K; \ v = 1,\ldots,m_k$, and $w = 1,\ldots,n_k$.

Therefore, $\text{prox}_{\eta\mathcal{L}}\left[\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}\right]$ can be represented by the proximal operator of each component $S_{k,v,w}(\cdot)$. Each additive component $S_{k,v,w}(\cdot)$ is a quadratic function of the elements in $\mathcal{B}_{k,v,w}^{(1)}$, whose proximal operator can be calculated using Proposition 1 given in Parikh and Boyd (2014). As each set $\mathcal{B}^{(1)}_{k,u,v}$ contains no more than $\sum_{i=1}^{K} q_i + 1$ parameters, the inversion of the matrix therein is performed rapidly with little difficulty.

Proposition 1. *If $q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$, with $\mathbf{A}$ being a positive semidefinite matrix:*

$$\text{prox}_{\eta q(\cdot)}(\mathbf{v}) = (\mathbf{I} + \eta\mathbf{A})^{-1}(\mathbf{v} - \eta\mathbf{b}), \tag{11}$$

where $\mathbf{I}$ is an identity matrix with the same size as $\mathbf{A}$.

The proximal operator of $p_1\left(\mathcal{B}^{(2)}_0, \mathcal{B}^{(2)}\right) + p_2\left(\mathcal{B}^{(2)}_0, \mathcal{B}^{(2)}\right)$. Note that

$$p_1\left(\mathcal{B}^{(2)}_0, \mathcal{B}^{(2)}\right) = \sum_{k\in\mathcal{S}}\left(\lambda_{1,k}\sum_{v=0}^{m_k}\|\mathbf{D}_S\mathbf{B}^{(2)}_{k0}(v,:)\|_2^2\right) + \sum_{k\in\mathcal{S}}\sum_{i=1}^{k}\sum_{j=1}^{q_i}\left(\lambda_{1,k}\sum_{v=1}^{m_k}\|\mathbf{D}_S\mathbf{B}^{(2)}_{ij,k}(v,:)\|_2^2\right),$$

$$p_2\left(\mathcal{B}^{(2)}_0,\mathcal{B}^{(2)}\right) = \sum_{k\in\mathcal{I}}\left(\lambda_{2,k}\text{vec}\left(\mathbf{B}^{(2)}_{k0}\right)^\top \mathbf{R}_I\text{vec}\left(\mathbf{B}^{(2)}_{k0}\right)\right) + \sum_{k\in\mathcal{I}}\sum_{i=1}^{k}\sum_{j=1}^{q_i}\left(\lambda_{2,k}\text{vec}\left(\mathbf{B}^{(2)}_{ij,k}\right)^\top \mathbf{R}_I\text{vec}\left(\mathbf{B}^{(2)}_{ij,k}\right)\right).$$

They are both the summation of multiple components involving disjoint sets of parameters. Each component corresponds to one parametric matrix, either an offset matrix $\mathbf{B}^{(2)}_{k0}$ or an effect matrix $\mathbf{B}^{(2)}_{ij,k}$. In the summations involving "$k \in \mathcal{S}$," every term is represented as $\lambda_{1,k}\sum_{v=0}^{m_k}\|\mathbf{D}_S\mathbf{T}(v,:)\|_2^2 := \sum_{v=1}^{m_k} p_S(\mathbf{T}(v,:))$ where $p_S(\mathbf{x}) = \lambda_{1,k}\|\mathbf{D}_S\mathbf{x}\|_2^2$. In the summation involving "$k \in \mathcal{I}$," every term is then represented as $p_I(\mathbf{T}) = \lambda_{2,k}\text{vec}(\mathbf{T})^\top \mathbf{R}_I\text{vec}(\mathbf{T})$. Therefore, it is sufficient to evaluate the proximal operator of $p_S(\mathbf{x})$ and $p_I(\mathbf{T})$, due to the separable property of the proximal operator (9). We derive the efficient evaluation of these proximal operators in Propositions 2 and 3. The proofs are given in Section A of the online supplement.

Proposition 2. *Given a d-dimensional signal $\mathbf{x} \in \mathbb{R}^d$, Algorithm 2 evaluates $\tilde{\mathbf{x}} = \text{prox}_{\eta p_S}(\mathbf{x})$, where $p_S(\mathbf{x}) = \lambda_1\|\mathbf{D}_S\mathbf{x}\|_2^2$.*

Proposition 3. *Given an $m \times n$ signal $\mathbf{T} = (t_{ij})_{m\times n} \in \mathbb{R}^{m\times n}$, Algorithm 3 evaluates $\tilde{\mathbf{T}} = \text{prox}_{\eta p_I}(\mathbf{T})$, where $p_I(\mathbf{T}) = \lambda_2\text{vec}(\mathbf{T})^\top \mathbf{R}_I\text{vec}(\mathbf{T})$.*

Algorithm 2. Calculate $\tilde{\mathbf{x}} = \text{prox}_{\eta p_S}(\mathbf{x})$

```
1: Calculate $\mathbf{x}^* = \text{DCT}(\mathbf{x})$, where DCT represents the 1D discrete cosine transform (Ahmed et al., 1974).
2: Set $\tilde{x}_i^* \leftarrow x_i^*/\left[1 + 4\lambda_1\eta\left(1 - \cos\left(\frac{i-1}{d}\pi\right)\right)^2\right]$ for $i = 1, 2, \ldots, d$, where $x_i^*$ is the $i$th element of $\mathbf{x}^*$.
3: Calculate $\tilde{\mathbf{x}} = \text{IDCT}(\tilde{\mathbf{x}}^*)$, where $\tilde{\mathbf{x}}^* = (\tilde{x}_1^*, \ldots, \tilde{x}_d^*)^\top$ and IDCT represents the inverse discrete cosine transform.
```

Algorithm 3. Calculate $\tilde{\mathbf{T}} = \text{prox}_{\eta p_I}(\mathbf{T})$

```
1: Calculate $\mathbf{T}^* \leftarrow \text{DCT2}(\mathbf{T})$, where DCT2 represents the 2D discrete cosine transform.
2: Set $\tilde{t}_{ij}^* \leftarrow t_{ij}^*/\left[1 + 4\lambda_2\eta\left(2 - \cos\left(\frac{i-1}{m}\pi\right) - \cos\left(\frac{j-1}{n}\pi\right)\right)^2\right]$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, where $t_{ij}^*$ is the $(i,j)$ element of $\mathbf{T}^*$.
3: Calculate $\tilde{\mathbf{T}} \leftarrow \text{IDCT2}(\tilde{\mathbf{T}}^*)$, where $\tilde{\mathbf{T}}^* = \left(\tilde{t}_{ij}^*\right)_{m\times n}$ and IDCT2 represents the inverse 2D discrete cosine transform.
```

The proximal operator of $p_3\left(\mathcal{B}^{(3)}\right)$. The penalty

$$p_3\left(\mathcal{B}^{(3)}\right) = \sum_{i=1}^{K}\left[\lambda_{3,i}\sum_{j=1}^{q_i}\|\mathbf{B}^{(3)}_{ij,\cdot}\|_2\right]$$

is also the summation of multiple components, each involving $\mathbf{B}^{(3)}_{ij,\cdot}$. By Equation (9), the proximal operator of $p_3(\cdot)$ can be evaluated by the proximal operators of each term $\lambda_{3,i}\|\mathbf{B}^{(3)}_{ij,\cdot}\|_2$, given in Proposition 4 (Parikh and Boyd, 2014).

Proposition 4.

$$\text{prox}_{\lambda\|\cdot\|_2}(\mathbf{x}) = \begin{cases}\left(1 - \dfrac{\lambda}{\|\mathbf{x}\|_2}\right)\mathbf{x}, \text{if } \|\mathbf{x}\|_2 \geq \lambda \\ \mathbf{0}, \text{if } \|\mathbf{x}\|_2 < \lambda\end{cases}.$$

The proximal operator of $p_4\left(\mathcal{B}^{(4)}\right)$. The separable property of the proximal operator (9) can be invoked again to calculate the proximal operator of

$$p_4\left(\mathcal{B}^{(4)}\right) = \sum_{k=1}^{K}\lambda_{4,k}\|\mathbf{B}^{(4)}_{\cdot\cdot,k}\|_*.$$

The closed-form expression for the proximal operator of $\lambda_{4,k}\|\cdot\|_*$ in Proposition 5 is also from Chapter 6.7.3 of Parikh and Boyd (2014). With this expression, the proximal operator of $p_4\left(\mathcal{B}^{(4)}\right)$ can be evaluated.

Proposition 5. Let A be an $m \times n$ matrix with singular value decomposition $\mathbf{A} = \sum_{i=1}^{\min\{m.n\}}\sigma_i\mathbf{u}_i\mathbf{v}_i^\top$. Then

$$\text{prox}_{\lambda_{4,k}\|\cdot\|_*}(\mathbf{A}) = \sum_{i=1}^{\min\{m,n\}}(\sigma_i - \lambda_{4,k})_+\mathbf{u}_i\mathbf{v}_i^\top,$$

where

$$x_+ = \begin{cases}x, x \geq 0 \\ 0, x < 0\end{cases}.$$

3.3.2. Evaluating the proximal operator in Step 2

According to Parikh and Boyd (2014), the proximal operator $\text{prox}_{\lambda g}[\cdot]$ involved in Step 2 is a projection onto the subspace $\left\{\tilde{\mathcal{B}} : \mathcal{B}^{(1)} = \mathcal{B}^{(2)} = \mathcal{B}^{(3)} = \mathcal{B}^{(4)}, \mathcal{B}^{(1)}_0 = \mathcal{B}^{(2)}_0\right\}$ and thus the update of $\tilde{\mathcal{Z}} = \left(\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \mathcal{Z}^{(3)}, \mathcal{Z}^{(4)}, \mathcal{Z}^{(1)}_0, \mathcal{Z}^{(2)}_0\right)$ is given by

$$\mathcal{Z}^{(i)} \leftarrow \overline{\mathcal{B}} + \overline{\mathcal{U}}, i = 1, 2, 3, \text{ and } 4; \mathcal{Z}^{(i)}_0 \leftarrow \overline{\mathcal{B}}_0 + \overline{\mathcal{U}}_0, i = 1 \text{ and } 2,$$

where $\overline{\mathcal{B}} := \frac{1}{4}\left(\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)} + \mathcal{B}^{(4)}\right)$, $\overline{\mathcal{B}}_0 := \frac{1}{2}\left(\mathcal{B}^{(1)}_0 + \mathcal{B}^{(2)}_0\right)$, and $\overline{\mathcal{U}}, \overline{\mathcal{U}}_0$ are defined accordingly. With the above specification of Step 2, $\overline{\mathcal{U}}, \overline{\mathcal{U}}_0$ will remain constant during iterations of the algorithm (though $\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(4)}$ and $\mathcal{U}^{(1)}_0$, $\mathcal{U}^{(2)}_0$ change during iterations), according to Step 3 of Algorithm 1. If $\overline{\mathcal{U}},\overline{\mathcal{U}}_0$ are initialized at zeros, the Step 2 of Algorithm 1 further reduces to $\mathcal{Z}^{(i)} \leftarrow \overline{\mathcal{B}}, \mathcal{Z}_0^{(i)} \leftarrow \overline{\mathcal{B}}_0$.

3.3.3. Summary of the proposed ADMM consensus algorithm

Now we put together the components listed in the above subsections and give a comprehensive optimization procedure in Algorithm 4. The notations $\overline{\mathcal{B}}_{k,v,w}, \overline{\mathbf{B}}_{k0}, \overline{\mathbf{B}}_{ij,k}, \overline{\mathcal{B}}_{i,j}$, and $\overline{\mathbf{B}}_{\cdot\cdot,k}$ are vectors or matrices, composed of elements in $(\overline{\mathcal{B}}, \overline{\mathcal{B}}_0)$, according to how $\mathcal{B}^{(1)}_{k,v,w}$ selects a subset of elements in $\left(\mathcal{B}^{(1)}, \mathcal{B}_0^{(1)}\right)$, how $\mathbf{B}^{(2)}_{k0}$ and $\mathbf{B}^{(2)}_{ij,k}$ select subsets of elements in $\left(\mathcal{B}^{(2)}, \mathcal{B}_0^{(2)}\right)$, how $\mathcal{B}^{(3)}_{i,j}$ selects a subset of elements in $\left(\mathcal{B}^{(3)}, \mathcal{B}_0^{(3)}\right)$, and how $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ selects a subset of elements in $\left(\mathcal{B}^{(4)}, \mathcal{B}_0^{(4)}\right)$, respectively. All notations involving the letter "U" corresponds to their counterparts involving the letter "B." For example, notation $\mathcal{U}^{(1)}_{k,v,w}$ in line (1a) refers to the subset of elements in $\mathcal{U}^{(1)}$, according to how $\mathcal{B}^{(1)}_{k,v,w}$ selects a subset of elements in $\mathcal{B}^{(1)}$.

Algorithm 4. The complete optimization procedure

```
Initiate $\overline{\mathcal{B}}=\mathcal{B}^{(1)}=\mathcal{B}^{(2)}=\mathcal{B}^{(3)}=\mathcal{B}^{(4)}=\mathcal{U}^{(1)}=\mathcal{U}^{(2)}=\mathcal{U}^{(3)}=\mathcal{U}^{(4)}=\mathcal{O}$ of the same shape as $\mathcal{B}$.
Initiate $\overline{\mathcal{B}}_0=\mathcal{B}_0^{(1)}=\mathcal{B}_0^{(2)}=\mathcal{U}_0^{(1)}=\mathcal{U}_0^{(2)}=\mathcal{O}$ of the same shape as $\mathcal{B}_0$.
Do:
(1) Save $\overline{\mathcal{B}}_{0,\ \mathrm{prev}} \leftarrow \overline{\mathcal{B}}_0$ and $\overline{\mathcal{B}}_{\mathrm{prev}} \leftarrow \overline{\mathcal{B}}$.
(2a) For $k = 1,\dots,K, v = 1,\dots,m_k, w = 1,\dots,n_k$ do:
    Update $\mathcal{B}^{(1)}_{k,v,w}$ by $\mathcal{B}^{(1)}_{k,v,w} \leftarrow \mathrm{prox}_{\eta S_{k,v,w}(\cdot)}\left(\overline{\mathcal{B}}_{k,v,w} - \mathcal{U}^{(1)}_{k,v,w}\right)$
    according to Equations (10) and (11).
(2b) For $k = 1,\dots,K, i = 1,\dots,k, j = 1,\dots,q_i$ do:
    If $k \in \mathcal{I}$: update $\mathbf{B}^{(2)}_{k0} \leftarrow \mathrm{prox}_{\eta p_I}\left(\overline{\mathbf{B}}_{k0} - \mathbf{U}^{(2)}_{k0}\right)$, $\mathbf{B}^{(2)}_{ij,k} \leftarrow \mathrm{prox}_{\eta p_I}\left(\overline{\mathbf{B}}_{ij,k} - \mathbf{U}^{(2)}_{ij,k}\right)$ based on Proposition 3.
    If $k \in \mathcal{S}$: update $\mathbf{B}^{(2)}_{k0}(v,:) \leftarrow \mathrm{prox}_{\eta p_S}\left(\overline{\mathbf{B}}_{k0}(v,:) - \mathbf{U}^{(2)}_{k0}(v,:)\right)$
    and $\mathbf{B}^{(2)}_{ij,k}(v,:) \leftarrow \mathrm{prox}_{\eta p_S}\left(\overline{\mathbf{B}}_{ij,k}(v,:) - \mathbf{U}^{(2)}_{ij,k}(v,:)\right)$ for all $v = 1,\dots,m_k$ based on Proposition 2.
(2c) For $i = 1,\dots,K$ and $j = 1,\dots,q_i$ do:
    Update $\mathbf{B}^{(3)}_{ij,\cdot} \leftarrow \mathrm{prox}_{\eta\lambda_{3,i}\|\cdot\|_2}\left(\overline{\mathbf{B}}_{ij,\cdot} - \mathbf{U}^{(3)}_{ij,\cdot}\right)$ based on Proposition 4.
(2d) For $k = 1,\dots,K$ do:
    Update $\mathbf{B}^{(4)}_{\cdot\cdot,k} \leftarrow \mathrm{prox}_{\eta\lambda_{4,k}\|\cdot\|_*}\left(\overline{\mathbf{B}}_{\cdot\cdot,k} - \mathbf{U}^{(4)}_{\cdot\cdot,k}\right)$ based on Proposition 5.
(3) Update $\overline{\mathcal{B}}$ and $\overline{\mathcal{B}}_0$ via $\overline{\mathcal{B}} \leftarrow \frac{1}{4}\left(\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)} + \mathcal{B}^{(4)}\right)$
    and $\overline{\mathcal{B}}_0 \leftarrow \frac{1}{2}\left(\mathcal{B}_0^{(1)} + \mathcal{B}_0^{(2)}\right)$.
(4) Update $\mathcal{U}^{(t)} \leftarrow \mathcal{U}^{(t)} + \mathcal{B}^{(t)} - \overline{\mathcal{B}}$ for $t = 1,\dots,4$ and
    $\mathcal{U}_0^{(t)} \leftarrow \mathcal{U}_0^{(t)} + \mathcal{B}_0^{(t)} - \overline{\mathcal{B}}_0$ for $t = 1,\ 2$.
Until $\max_{i=1,\dots,4}\|\overline{\mathcal{B}} - \mathcal{B}^{(i)}\|$, $\max_{i=1,2}\|\overline{\mathcal{B}}_0 - \mathcal{B}_0^{(i)}\|$, $\max_{i=1,\dots,4}\|\overline{\mathcal{B}} - \overline{\mathcal{B}}_{\mathrm{prev}}\|$
and $\max_{i=1,2}\|\overline{\mathcal{B}}_0 - \overline{\mathcal{B}}_{0,\ \mathrm{prev}}\|$ are below $\epsilon$.
```

In Algorithm 4, all updating operations within the four "for loops" in step (2a)-(2d) can be performed in parallel, as they involve distinct groups of elements in $\tilde{\mathcal{B}}$. This notable feature significantly improves the computational efficiency. The variables in the optimization problem include the offset matrices $\mathcal{B}_0$ and the effect matrices $\mathcal{B}$ listed in the cells of Table 1, which contains $\sum_{k=1}^K q_k$ columns and K rows. In essence, the step (2a)-(2d) of the algorithms updates $\left(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}\right)$, $\left(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}\right)$, $\mathcal{B}^{(3)}$ and $\mathcal{B}^{(4)}$ through operating on multiple groups of elements in parallel. In step (2a), $\left(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}\right)$ is divided into $\sum_{k=1}^K m_k n_k$ groups. Each group corresponds to a triple (k,v,w) where $k \in \{1,\dots,K\}, v \in \{1,\dots,m_k\}$ and $w \in \{1,\dots,n_k\}$, and it consists of the (v,w)-element of $\mathbf{B}_{k0}$ and all (v,w)-element of matrices listed in the kth row of Table 1. In step (2b), $\left(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}\right)$ is updated by breaking them into $\sum_{i=1}^K (k+1-i)q_i + K$ groups according to the *cells* of Table 1 and the matrices in $\mathcal{B}_0^{(2)}$. In step (2c), $\mathcal{B}^{(3)}$ is divided according to $\sum_{i=1}^K q_i$ columns of Table 1. In step (2d), $B^{(4)}$ is divided according to K rows of Table 1.

In Section 3.1.1, we discussed how to answer the questions of the diagnostics based on the estimation of $\mathcal{B}, \mathcal{B}_0$. When the algorithm terminates, however, such sparsity and low-rank property cannot be observed from $\overline{\mathcal{B}}$, due to the numerical error. However, $\mathbf{B}^{(3)}_{ij,\cdot}$ can be all zeros for many pairs of (i,j) given appropriate values of tuning parameters, as it is obtained from the proximal operator for an ℓ_2-norm. Therefore, we may identify whether each u_{ij} affects the output quality measurements by observing whether $\mathbf{B}^{(3)}_{ij,\cdot} = \mathbf{O}$. Similarly, the $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ is of low-rank, and its rank specifies the number of variation patterns on stage k that all inputs cause. Finally, $\mathbf{B}_{ij,k}$, the effect of the input u_{ij} on stage k can be visualized through heat maps or multiple curves corresponding to $\mathcal{B}^{(2)}$, to give a visualization of smooth curves or images. As the algorithm converges, note that the difference between $\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(4)}$ is very small, and all of them are close to $\overline{\mathcal{B}}$.

The convergence of the ADMM algorithm is guaranteed in the literature (Boyd *et al.*, 2011). However, the existing theory on the convergence rate of a consensus ADMM algorithm relies on the strong convexity assumption of component functions, which does not hold for $p_2(\mathcal{B}_0, \mathcal{B})$, $p_3(\mathcal{B})$ and $p_4(\mathcal{B})$. Section B of the online supplement gives a comprehensive analysis of the computation complexity of each iteration of the algorithm. We leave the illustration of the empirical convergence behavior of the algorithm in the simulation study.

3.4. Discussion

Here we discuss the selection of the tuning parameters and possible variations of the problem formulations.

Table 1. Effect matrices estimated in formulation (7).

$\mathbf{B}_{11,1}$	$\cdots$	$\mathbf{B}_{1q_1,1}$							
$\mathbf{B}_{11,2}$	$\cdots$	$\mathbf{B}_{1q_1,2}$	$\mathbf{B}_{21,2}$	$\cdots$	$\mathbf{B}_{2q_2,2}$				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$			
$\mathbf{B}_{11,K}$	$\cdots$	$\mathbf{B}_{1q_1,K}$	$\mathbf{B}_{21,K}$	$\dots$	$\mathbf{B}_{2q_2,K}$	$\dots\dots$	$\mathbf{B}_{K1,K}$	$\dots$	$\mathbf{B}_{Kq_k,K}$

3.4.1. Selection of tuning parameters

In our formulation (7), the values of the tuning parameters $\{\lambda_{1,k},\ldots,\lambda_{4,k}\}$ need to be specified. The literature often suggests setting the tuning parameters through a Cross-Validation (CV) procedure, although our simulation study shows that it under-smoothes the signals and the images, and leads to a larger number of effective parameters and variation patterns. The same finding was highlighted by Yan *et al.* (2017), and they adopted Otsu's method based on maximizing inter-class variance. However, this method cannot be extended in our application, as classes are not well-defined.

In general, a large value of $\lambda_{1,k}$ leads to smoother response signals of stage k, and a large value $\lambda_{2,k}$ leads to smoother image responses of stage k. A larger value of $\lambda_{3,i}$ leads to a fewer number of effective inputs from stage i, and larger value of $\lambda_{4,k}$ leads to a fewer number of variation patterns in stage k. According to the algorithm in Section 3.3, $\lambda_{4,k}$ aims to threshold the singular values of the matrix $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ of size $m_k n_k \times \sum_{i=1}^{k} q_i$, and the variance of the error of every entry of $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ is proportional to $\sigma^2_{E,k}$. According to Yuan *et al.* (2007), if an $p\times q$ matrix is the summation of a rank-r matrix and a matrix of $N(0,\sigma^2)$ error, the $(r+1)$th singular value is of magnitude $\sigma\left(\sqrt{p}+\sqrt{q}\right)$. It motivates us to take

$$\lambda_{4,k} = c_{4,k}\sigma_{E,k}\left(\sqrt{m_k n_k} + \sqrt{\sum\nolimits_{i=1}^{k} q_i}\right),$$

where $c_{4,k}$ is a prescribed constant, to make the magnitudes of shrinkage applied on the matrices $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ comparable, even if their shapes differ.

The selection of $\lambda_{3,i}$ and $c_{4,k}$ should be regarded as a decision driven by engineering need. For example, if the practitioners solve the model for identifying a wide range of inputs and output variations for root cause diagnosis, $\lambda_{3,i}$ and $c_{4,k}$ should be set to smaller values, which will lead to identifying more effective inputs and variation patterns. If the practitioners are only interested in the inputs that have major effects on the output variation patterns, larger values of $\lambda_{3,i}$ and $c_{4,k}$ are preferable.

Finally, we note that depending on an actual physical system, the tuning parameters $\{\lambda_{1,k},\lambda_{2,k},\lambda_{3,k},c_{4,k}: k=1,\ldots,K\}$ corresponding to similar stages may be divided into multiple groups. Each group of parameters can take the same values, or be selected using the same policy to reduce the complexity, as illustrated in the simulation study.

3.4.2. Variation of problem formulations based on process specifications

We finally note that the analytical framework presented in this section can be extended and configured based on the specific layout of the MMP and sensing system. First, some processes generate both functional curves and images in certain stages or generate curves and images of different sizes. The problem formulation and optimization algorithm can be applied with some minor modifications. Second, if the curves and images have various smoothness properties, different roughness penalties may be applied by discretizing the roughness penalties for functional data (see Section 5.3.3 of Ramsay (2005)). Third, certain manufacturing stages generate other forms of data, such as the spatial measurements seen in lithography processes, point cloud data in machining processes, as well as electrical signals that jump at discrete time points. Associated penalties based on spatial coordinates and point distance should be applied based on the structure of such data, instead of using smoothness penalties presented above. The ADMM consensus algorithm can be adjusted accordingly with minor modifications.

As a limitation of the proposed model, we assumed a linear relationship between the process inputs and the process outputs. If their relationship is not linear, we may include quadratic terms of the input variables into the model. We can also transform the process outputs from one stage into a set of meaningful features that are linearly related to the inputs. Furthermore, the interaction effect of two or more process inputs can be studied in our analytical framework as well, by including $u_i u_j$ terms in the inputs.

Finally, we note that the least square loss function can be replaced with other types of loss functions, such as Huber or Tukey loss, to yield more robust solutions when the error follows heavy-tail distributions or when outliers exist. If the loss function is convex, and its proximal operator can be evaluated effectively, the ADMM consensus framework can still be applied.

4. Simulation studies for performance evaluation

In this section, we set up a simulation platform to validate the methodology proposed in Section 3. We will demonstrate how to use the proposed framework to specify the number of variation patterns and effective inputs from the process inputs and intermediate product quality measurements.

4.1. Engineering background

The manufacturers of semiconductors are interested in discovering how process inputs relate to the intermediate product quality measurements. However, there are no effective

Table 2. The type and dimension of data from each stage.

Stage	*1*	*2*	*3*	*4*
Data type	multiple functional signals	multiple functional signals	images	images
Output dimension	3 signals of length 10	4 signals of length 20	10×10	20×20
Input dimension	$q_1 = 20$	$q_2 = 20$	$q_3 = 20$	$q_4 = 20$

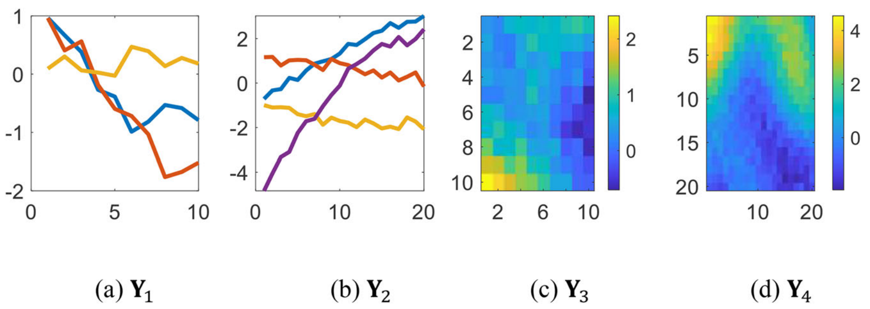

Figure 2. The outputs of curves and images from four stages.

methods to identify the relationships between them. The characteristics of the semiconductor manufacturing process discussed in the Introduction motivates our simulation setup. In practice, only a small number of variation patterns in outputs present in a given time period. This is because each variation pattern of quality data is typically driven by a few root causes associated with process inputs. A well-maintained process should have limited variation sources in a short period. As we introduced in Section 3.1, the number of variation patterns for the outputs from stage k is represented by the rank of the parametric matrix $\mathbf{B}_{\cdot\cdot,k}$. As only a small portion of the inputs affect the quality data, estimation of the effect matrices can be cast as a low rank, sparse estimation problem.

In our simulation study, we: (i) evaluate whether the offset matrices $\{\mathbf{B}_{k0}: k = 1, ..., K\}$ and the effect matrices $\{\mathbf{B}_{ij,k}\}$ can be estimated accurately; (ii) identify the inputs related to the outputs; and (iii) find the number of input-driven variation patterns presented in the outputs. The detailed setup of the simulation study is described in Section 4.2, and the results are discussed in Section 4.3.

4.2. Specifications of simulation settings

Our simulation testbed has a total of $K = 4$ stages. Each involves $q_i = 20$ input variables. The type and dimension of data generated from each stage are summarized in Table 2. Our simulation is performed under a relatively low dimensional setting compared with real applications, as it enables us to conduct the simulation thoroughly with more replications and under a wider variety of problem settings.

We aim at simulating the system that satisfies the following conditions: (i) r_k variation patterns present in the outputs of stage k, (ii) only the q_k effective inputs from stage 1 to k relates to the quality measurements of stage k, and (iii) the image or multiple functional quality measurements are smooth. We generate the true values of $\mathcal{B}_0$ and $\mathcal{B}$, so that: (i) the matrix of $\mathbf{B}_{\cdot\cdot,k}$ has a rank of r_k (unless the number of rows or columns of $\mathbf{B}_{\cdot\cdot,k}$ is smaller than r_k); (ii) $\mathbf{B}_{ij,k} \neq \mathbf{O}$ if and only if u_{ij} is an effective parameter; and (iii) the rows of $\mathbf{B}_{k0}$ and $\mathbf{B}_{ij,k}$ form smooth curves if $k \in \mathcal{S}$, and the matrices $\mathbf{B}_{k0}$ and $\mathbf{B}_{ij,k}$ form smooth images if $k \in \mathcal{J}$. For every stage $k = 1, ..., K$, we first generate $\mathbf{B}_{k0}$ using multiple univariate Gaussian processes (if $k \in \mathcal{S}$) or a bivariate Gaussian process (if $k \in \mathcal{J}$). Then we generate r_k basis, using the same procedure of generating $\mathbf{B}_{k0}$. We finally generate $\mathbf{B}_{ij,k}$ corresponding to all effective input u_{ij} using a random linear combination of these r_k basis, such that the rank of $\mathbf{B}_{\cdot\cdot,k}$ is r_k. The detailed procedure of generating $\mathcal{B}_0$ and $\mathcal{B}$ is given in Section C of the online supplement. Given $\mathcal{B}$ and $\mathcal{B}_0$, we then generate the data corresponding to $N = 500$ products. The process inputs u_{ij} for each product are independent standard normal random variables, and the process outputs of each product are generated according to model (1) based on the process inputs, where $\mathbf{E}_k^{\{n\}}$ are independent standard normal random variables with variance $\sigma_E^2 = 0.2$. Figure 2 illustrates the output data collected from one sample, where the four subfigures are the multiple functional signals and image data collected from each manufacturing stage.

In our simulation studies, we consider the four process setups in which the number of potential root causes from each stage (reflected by r_k, the rank of $\mathbf{B}_{\cdot\cdot,k}$, $k = 1, ...K$) and the number of effective process inputs from each stage ($q_{e,k}, k = 1, ..., K$) are varied: (i) $r_k = 2; q_{e,k} = 3$; (ii) $r_k = 5$; $q_{e,k} = 3$; (iii) $r_k = 2$; $q_{e,k} = 6$ and (iv) $r_k = 5$; $q_{e,k} = 6$ for all $k = 1, ..., K$. For each simulation setting, we perform the estimation for 300 times, according to 30 different pairs of offset matrices $\mathcal{B}_0$ and effect matrices $\mathcal{B}$ that define 30 specific manufacturing process. For each pair, we generated 10 datasets corresponding to different inputs $\mathbf{u}_1, ..., \mathbf{u}_K$ and

Table 3. Summary of the simulation settings.

- Specifies r_k and $q_{e,k}$ according to Setting (1)-(4).
  - Given each specification of $\{r_k, q_{e,k} : k = 1, \ldots, K\}$, generate 30 sets of $\{\mathcal{B}_0, \mathcal{B}\}$ according to the procedure detailed in Section C of the online supplement.
    - Generate 10 sets of inputs $\{\mathbf{u}_1, \ldots, \mathbf{u}_K\}$ whose elements are all independent and follow standard normal distributions, and generate the error $\{\mathbf{E}_1, \ldots, \mathbf{E}_K\}$ whose elements are all independent and follow $N\left(0, \sigma^2_{E,k}\right)$, $k = 1, \ldots, K$.
    - Given each specification of $\{\mathcal{B}_0, \mathcal{B}\}$, the set of $\{\mathbf{u}_1, \ldots, \mathbf{u}_K\}$ and the error $\{\mathbf{E}_1, \ldots, \mathbf{E}_K\}$, simulate 10 datasets, each contains $\mathcal{Y}^{\{n\}} = \left(\mathbf{Y}_1^{\{n\}}, \ldots, \mathbf{Y}_K^{\{n\}}\right), n = 1, \ldots, N$, based on Equation (1). The sample size $N = 500$.
      - From each dataset, estimate $\hat{\mathcal{B}}_0$ and $\hat{\mathcal{B}}$.

random errors $\mathbf{E}_1 \ldots, \mathbf{E}_K$, representing process inputs and outputs collected from the same process. These simulation settings are summarized in Table 3.

4.3. Optimization procedure and results of simulation studies

Based on each generated dataset with sample size $N = 500$, we perform the modeling and estimation procedure described in Section 3. Although the number of effective inputs $q_{e,k}$ and the number of variation patterns r_k differs, we select the same set of tuning parameters: $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 0.25$ and $\lambda_4^k = 0.2\left(\sqrt{kJ} + \sqrt{m_k n_k}\right), k = 1, \ldots, K$ based on the discussion in Section 3.4.1. We fix the step size $\eta = 5$ under all simulation settings.

Our algorithm is implemented in MATLAB, and the simulation study is conducted on a computing cluster. We did not implement the parallel for-loops in the algorithm in a parallel computing framework. To illustrate the speed and convergence property of the algorithm, we perform the estimation for one dataset, generated with $r_k = 2$ and $q_{e,k} = 3$, on a standalone mobile workstation with Intel Xeon E-2176M 2.7GHz CPU and 16GB memory. The stopping criterion is that both the primal residual

$$\epsilon_{\text{prim}} = \max\left\{\max_{i=1,\ldots,4} \|\overline{\mathcal{B}} - \mathcal{B}^{(i)}\|, \max_{i=1,2} \|\overline{\mathcal{B}}_0 - \mathcal{B}_0^{(i)}\|\right\}$$

and the dual residual

$$\epsilon_{\text{dual}} = \max\left\{\max_{i=1,\ldots,4} \|\overline{\mathcal{B}} - \overline{\mathcal{B}}_{\text{prev}}\|, \max_{i=1,2} \|\overline{\mathcal{B}}_0 - \overline{\mathcal{B}}_{0,\ \text{prev}}\|\right\}$$

are below 10^{-5}, where $\|\cdot\|$ is the max norm. The algorithm converges in 2133 iterations. On average, each iteration takes 0.61 s. We observed that the major computational burden is in step (2a), where each iteration takes an average of 0.51 s. In step (2a) we need to construct and solve $m_1 n_1 = 30$ linear systems of order 20, 80 linear systems of order 40, 100 linear systems of order 60, and 400 linear systems of order 80. However, these linear systems could be solved in parallel. As for the convergence speed, the change of $\log \epsilon_{\text{prim}}$ and $\log \epsilon_{\text{dual}}$ in all iterations are illustrated in Figure 3. From this figure, we can see that the primal and dual residuals are consistently dropping. However, the algorithm has a sublinear convergence rate.

After the estimation of $\mathcal{B}_0, \mathcal{B}$ are obtained, we observe that the estimation of $\hat{\mathbf{B}}_{ij,k}$ and $\hat{\mathbf{B}}_{k0}$ are either multiple smooth curves when $k = 1,\ 2$ or smooth images when $k = 3,\ 4,$ which are consistent with the true system parameters (see Figure 4 for the illustration of the estimation of $\hat{\mathbf{B}}_{10}, \ldots, \hat{\mathbf{B}}_{40},$ for example). All estimation of $\hat{\mathbf{B}}_{ij,k}$ in one run is shown in Section D of the online supplement. We can see that the estimated parametric matrices satisfy the corresponding smoothness property.

We then evaluate the effectiveness of the root cause analysis based on the estimations. Specifically, we: (i) evaluate their estimation accuracies based on the difference between each estimated parametric matrix $\hat{\mathbf{B}}_{ij,k}$ or $\hat{\mathbf{B}}_{k0}$ and their corresponding true value, $\mathbf{B}_{ij,k}$ or $\mathbf{B}_{k0}$; (ii) identify the effective inputs from each stage by checking whether the entries associated with each u_{ij} of parameter set $\mathbf{B}^{(3)}_{ij,\cdot}$ is non-zero; and (iii) identify the number of variation patterns of each stage through the number of positive singular values of the estimated $\hat{\mathbf{B}}^{(4)}_{\cdot\cdot,k}$ (Parikh and Boyd, 2014).

4.3.1. The estimation accuracy

From the estimation $\hat{\mathbf{B}}_{ij,k}$ obtained from each dataset, we calculate:

$$d_{i,k} = \sqrt{\frac{1}{q_i m_k n_k} \sum\nolimits_{j=1}^{q_i} \|\mathbf{B}_{ij,k} - \hat{\mathbf{B}}_{ij,k}\|_F^2},$$

and

$$d_{k0} = \sqrt{\frac{1}{m_k n_k} \|\mathbf{B}_{k0} - \hat{\mathbf{B}}_{k0}\|_F^2},$$

to evaluate the estimation error. These quantities correspond to the rooted mean square error associated with every element of the estimated matrices. Under settings (i) to (iv), let $d_{i,k}(n_{\text{par}}, n_{\text{rep}})$ and $d_{k0}(n_{\text{par}}, n_{\text{rep}})$ be the values of $d_{i,k}$ and d_{k0} calculated from the dataset according to the n_{par}-th generation of $\{\mathcal{B}_0, \mathcal{B}\}$ and n_{rep}-th generation of inputs and random errors $(n_{\text{par}} \in \{1, \ldots, 30\}, n_{\text{rep}} \in \{1, \ldots, 10\})$. To understand the average estimation accuracy within each setting and the uncertainty of the estimation error, we further calculate the following summary statistics:

1. The average error of the setting $\hat{\mu} = \hat{\mathrm{E}}_{n_{\text{par}}} \hat{\mathrm{E}}_{n_{\text{rep}}} \left[d_{i,k}(n_{\text{par}},\ n_{\text{rep}})\right]$.
2. The variability of the error caused by inputs and random error uncertainty in replications:
$$\hat{\sigma}_{\text{rep}} = \sqrt{\hat{\mathrm{E}}_{n_{\text{par}}} \widehat{\text{var}}_{n_{\text{rep}}} \left[d_{i,k}(n_{\text{par}}, n_{\text{rep}})\right]}.$$
3. The variability of the error caused by different generations of parameters $\{\mathcal{B}_0,\ \mathcal{B}\}$

Table 4. The estimation error $d^2_{i,k}(1 \le i \le k \le 4)$ and the associated σ_{rep} and σ_{par} in brackets.

	$k=1$	$k=2$	$k=3$	$k=4$
Setup 1: $r_k = 2,\ q_{e,k} = 3$				
$i=1$	1.36e-04 (2.03e-10 / 2.63e-10)	1.37e-04 (8.39e-09 / 4.57e-09)	1.41e-04 (2.87e-08 / 6.98e-08)	2.78e-04 (5.42e-06 / 2.71e-06)
$i=2$		1.37e-04 (1.10e-08 / 7.70e-09)	1.45e-04 (1.74e-07 / 1.40e-07)	6.18e-04 (1.81e-05 / 1.12e-05)
$i=3$			1.54e-04 (7.16e-07 / 3.33e-07)	3.39e-03 (1.95e-04 / 8.61e-05)
$i=4$				1.14e-01 (2.57e-03 / 6.59e-03)
Setup 2: $r_k = 5,\ q_{e,k} = 3$				
$i=1$	1.36e-04 (2.77e-10 / 1.73e-10)	1.37e-04 (3.67e-09 / 2.83e-09)	1.41e-04 (2.43e-07 / 4.81e-08)	2.83e-04 (5.06e-06 / 1.84e-06)
$i=2$		1.37e-04 (1.15e-08 / 6.93e-09)	1.45e-04 (1.75e-07 / 1.31e-07)	6.67e-04 (2.00e-05 / 9.90e-06)
$i=3$			1.56e-04 (4.27e-07 / 2.01e-07)	3.46e-03 (2.64e-04 / 7.32e-05)
$i=4$				1.16e-01 (8.28e-03 / 5.54e-03)
Setup 3: $r_k = 2,\ q_{e,k} = 6$				
$i=1$	1.36e-04 (2.23e-10 / 2.84e-10)	1.37e-04 (3.67e-09 / 2.91e-09)	1.41e-04 (1.09e-07 / 9.50e-08)	2.89e-04 (3.49e-06 / 2.35e-06)
$i=2$		1.37e-04 (7.48e-09 / 6.30e-09)	1.45e-04 (2.31e-07 / 1.78e-07)	6.59e-04 (3.27e-05 / 1.26e-05)
$i=3$			1.56e-04 (5.69e-07 / 2.77e-07)	3.74e-03 (1.81e-04 / 1.10e-04)
$i=4$				1.46e-01 (7.72e-03 / 8.26e-03)
Setup 4: $r_k = 5,\ q_{e,k} = 6$				
$i=1$	1.36e-04 (1.88e-10 / 1.65e-10)	1.37e-04 (2.56e-09 / 2.92e-09)	1.41e-04 (1.90e-07 / 4.65e-08)	2.93e-04 (1.45e-06 / 1.43e-06)
$i=2$		1.37e-04 (8.59e-09 / 5.00e-09)	1.46e-04 (2.46e-07 / 9.88e-08)	6.76e-04 (2.53e-05 / 8.24e-06)
$i=3$			1.57e-04 (5.00e-07 / 2.29e-07)	3.92e-03 (2.74e-04 / 7.30e-05)
$i=4$				1.52e-01 (5.43e-03 / 6.03e-03)

$$\hat{\sigma}_{\text{par}} = \sqrt{\widehat{\text{var}}_{n_{\text{par}}} \hat{\text{E}}_{n_{\text{rep}}}\left[d_{i,k}(n_{\text{par}}, n_{\text{rep}})\right]},$$

where $\hat{\text{E}}_{n_{\text{rep}}}$, $\hat{\text{E}}_{n_{\text{par}}}$ denotes the average of the following expression for $n_{\text{rep}} = 1, ..., 10$ or $n_{\text{par}} = 1, ..., 30$ respectively, and $\widehat{\text{var}}_{n_{\text{rep}}}$, $\widehat{\text{var}}_{n_{\text{par}}}$ denotes the sample variance of the following expression for $n_{\text{rep}} = 1, ..., 10$ or $n_{\text{par}} = 1, ..., 30$ respectively. The summarizing statistics of $\hat{\mu}, \hat{\sigma}_{\text{rep}}$ and $\hat{\sigma}_{\text{par}}$ of all $d_{i,k}$ in the four system settings are reported in Table 4 and Table 5, respectively. In each cell of this table, the number outside the bracket is the value of $\hat{\mu}$ corresponding to setup $1, ..., 4$, and the two numbers separated by the slash in each bracket are $\hat{\sigma}_{\text{rep}}$ and $\hat{\sigma}_{\text{par}}$ that respectively quantifies the uncertainty caused by the inputs and error, and the uncertainty caused by different generations of $\{\mathcal{B}, \mathcal{B}_0\}$.

From the results in the table, we can observe that the errors are small in general. We summarize the following findings:

1. Among all parametric matrices, the effects of the inputs from stage 1 on the output of stage 1 are estimated most accurately. The magnitude of the error is in the order of 10^{-4}. The reason is that the outputs from stage 1 are only related to the inputs from stage 1, and therefore the relationship between them is clear. Also, the smoothness penalty regularized the matrices of estimation, and thus increase the estimation accuracy.
2. As k increases from 1 to 4, the error associated with the estimation $\hat{\mathbf{B}}_{ij,k}$ generally increases. One of the reasons is that the total number of elements in $\mathbf{B}_{ij,k}$ increases. (Note that when $k = 1$, 2, 3 and 4, the number of elements in $\mathbf{B}_{ij,k}$ is respectively 30, 80, 100, 400). Apart from this, outputs from later stages are associated with more input variables, and thus the estimation accuracy decreases. Finally, later stages involve larger penalization due to the ranks, as $\lambda_{4,4} > \lambda_{4,3} > \lambda_{4,2} > \lambda_{4,1}$ in our setup. Although such selection of the hyper-parameters is necessary to reduce the ranks of $\mathbf{B}_{\cdot\cdot,k}$ involving later stages k containing more elements and associated inputs, it also introduces larger biases for the estimations in later stages.
3. When k is fixed, the estimation of $\mathbf{B}_{ij,k}$ become less accurate when i increases.
4. Among the four settings, the estimation is more accurate when $r = 2$ and less accurate when $r = 5$. The estimation is also more accurate when $q_e = 3$ and less accurate when $q_e = 6$. This is because our penalization works best when the number of variation patterns of each stage and the number of effective inputs from each stage is not high.

Except for the observations above, we can also see that $\hat{\sigma}_{\text{rep}}, \hat{\sigma}_{\text{par}}$ are typically much smaller than the error $\hat{\mu}$, which indicates that the uncertainty of the estimation error is not high and that the discoveries above are conclusive instead of merely out of chance.

4.3.2. Effective inputs

Among simulation setups 1 to 4, the number of effective inputs from each stage is either three or six. Of $30 \times 10 = 300$ replicates corresponding to the four setups, all inputs are correctly identified as effective or ineffective ones from stages 1, 2, and 3. In stage 4, on average 0.28 ineffective inputs are falsely selected as effective one under setup 2, 0.18 ineffective inputs are falsely selected as effective one under setup 3, and 0.28 ineffective inputs are falsely selected as effective one under setup 4. This result indicates that the proposed framework generally identifies the significant variables in early stages if the sample size is large enough, the error is not so big, and the hyperparameters are appropriately chosen. However, in later stages like stage 4, the proposed framework is likely to select extra ineffective inputs, because the total number of inputs that may affect this stage is too big (including all input from the current stage and the previous stages). Consequently, the algorithm is more prone to selecting ineffective variables.

Table 5. The estimation error of $d_{k0}^2(1 \le k \le 4)$ and the associated σ_{rep} and σ_{par} in brackets.

Setup 1: $r = 2,\ q_e = 3$			
5.82e-03 (2.38e-04 / 1.12e-03)	6.11e-03 (7.79e-04 / 1.87e-03)	9.58e-03 (1.48e-03 / 4.56e-03)	3.17e-02 (2.74e-03 / 1.58e-02)
Setup 2: $r = 5,\ q_e = 3$			
5.94e-03 (4.40e-04 / 1.06e-03)	8.12e-03 (8.31e-04 / 2.93e-03)	1.02e-02 (1.02e-03 / 3.48e-03)	3.49e-02 (2.98e-03 / 1.32e-02)
Setup 3: $r = 2,\ q_e = 6$			
6.11e-03 (4.75e-04 / 9.40e-04)	7.57e-03 (8.80e-04 / 2.60e-03)	8.88e-03 (1.34e-03 / 4.35e-03)	3.57e-02 (3.46e-03 / 1.92e-02)
Setup 4: $r = 5,\ q_e = 6$			
6.05e-03 (2.18e-04 / 9.78e-04)	8.70e-03 (6.42e-04 / 2.13e-03)	1.31e-02 (1.02e-03 / 5.18e-03)	4.36e-02 (6.24e-03 / 1.92e-02)

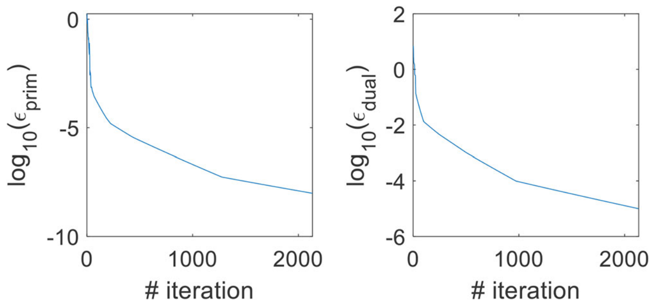

Figure 3. The convergence of ϵ_{prim} and ϵ_{dual} in all iterations.

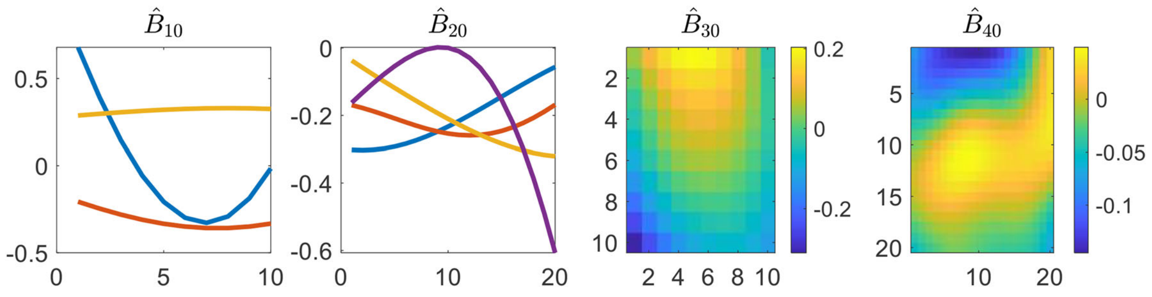

Figure 4. An illustration of an estimation of $\hat{\mathbf{B}}_{10}, \hat{\mathbf{B}}_{20}, \hat{\mathbf{B}}_{10}$ and $\hat{\mathbf{B}}_{40}$ from one dataset.

Table 6. Number of variation patterns and the associated σ_{rep} and σ_{par} in brackets.

Setup 1: $r_k = 2,\ q_{e,k} = 3$			
2 (0 / 0)	2 (0 / 0)	2 (0 / 0)	2 (0 / 0)
Setup 2: $r_k = 5,\ q_{e,k} = 3$			
3 (0 / 0)	4.60 (0.548 / 0)	4.38 (0.522 / 0.141)	4.42 (0.814 / 0.287)
Setup 3: $r_k = 2,\ q_{e,k} = 6$			
2 (0 / 0)	2 (0 / 0)	2 (0 / 0)	2 (0 / 0)
Setup 4: $r_k = 5,\ q_{e,k} = 6$			
4.60 (0.548 / 0)	5 (0 / 0)	4.80 (0.447 / 0)	5 (0 / 0)

4.3.3. Number of variation patterns

Within four simulation setups, the number of variation patterns from each stage is either two or five Within each simulation setup, we calculate $r(n_{\text{par}}, n_{\text{rep}})$, the rank of $\hat{\mathbf{B}}^{(4)}_{\cdot\cdot,k}$ corresponding to the n_{par} th generation of $\{\mathcal{B}, \mathcal{B}_0\}$ and the n_{rep} th generation of inputs and errors ($n_{\text{par}} \in \{1, ..., 30\}, n_{\text{rep}} \in \{1, ..., 10\}$). Similar to the report of estimation accuracy, the average rank among 300 simulation cases ($\hat{\mu}_r$) and their uncertainties $\hat{\sigma}_{\text{par,r}}, \hat{\sigma}_{\text{rep,r}}$ are reported in Table 6.

From the result, we observe that when the true number of variation patterns for the output in each stage is $r = 2$, the algorithm can always correctly identify two variation patterns (because $\hat{\sigma}_{\text{par}} = \hat{\sigma}_{\text{rep}} = 0$ indicates that $r(n_{\text{par}}, n_{\text{rep}})$ are the same for all $n_{\text{par}}, n_{\text{rep}} = 1, ..., 10$). However, in setup 2 where $q_{e,k} = 3, r_k = 5$, the average rank of $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ is 3, 4.60, 4.38, and 4.42 for stages 1, 2, 3, and 4, according to Table 6. The number of the variation patterns for the output from stage 1 is correctly given, as this stage is only affected by three inputs from itself, and the number of variation patterns cannot exceed three. The later stages 2, 3, and 4 are influenced by 6, 9, and 12 inputs, respectively, but the effects of them are restricted in a five-dimensional subspace. However, the algorithm does not always identify the rank as five. We guess that the reason is two-fold: (i) collinearity may exist among the randomly generated five variation patterns, and (ii) the number of inputs is small to reveal all variation patterns. Under setup 4, $q_{e,k} = 6$ and thus the number of inputs is larger. As a result, all replications in setup 4 identify that rank of $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ is five and thus the number of the variation patterns is correctly estimated in stages 2 and 4. In stages 1 and 3, the average rank of $\mathbf{B}^{(4)}_{\cdot\cdot,k}$ are 4.60 and 4.80 across all replications. Although error exists, the estimation is closer to the correct value five than the results in setup 2.

5. Conclusion

In a data-rich manufacturing environment, an MMP generates various types of data from different manufacturing stages, which poses a great challenge for data analytics. In this article, we propose a novel root cause diagnostic framework for an MMP that satisfies four assumptions: (i) the input from one stage only affect the down-stream stages (e.g., no re-work); (ii) the process outputs satisfies smoothness properties; (iii) only a small number of inputs affect the process outputs; and (iv) the variation patterns caused by the inputs are limited. Based on these assumptions, our approach identifies the effective inputs that relate to the perturbation of the outputs, identifies the variation patterns of the outputs caused by these inputs, and determines how each individual process input affects the manufacturing process.

The root cause diagnostic framework is based on a model for MMPs that generates mixed profile data, such as functional signals or images. For estimating the model parameters, we proposed a distributed computational scheme. The framework proposed in this article is highly extendable: the practitioners may customize it based on the special characteristics of the process, by using appropriate loss functions and structural assumptions on various types of data generated from different stages.

We developed the simulation study based on the scenario of a real semiconductor manufacturing process. In general, with correctly specified tuning parameters, the proposed method can perform well for three tasks of root cause diagnosis: it achieves satisfactory estimation accuracy, can correctly identify the inputs that affect the outputs, and provide a good estimation of the number of variation patterns for the output from each stage.

In this study, our modeling of the MMP focus on the application of root cause diagnosis. How to extend this modeling technique to process control, optimization, and sensor allocation are follow-up questions that need to be studied in the future. We will also extend our current framework to tensor inputs and outputs generated from each stage and modeling the inter-relationship between heterogenous intermediate quality measurement.

Notes on contributors

Andi Wang is a PhD student in the Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. He received his BS in statistics from Peking University in 2012 and a Ph.D. from Hong Kong University of Science and Technology in 2016. His research interests include advanced statistical modeling, large-scale optimization, and machine learning for manufacturing and healthcare system performance improvements via process monitoring, diagnostics, prognostics, and control. He is a member of IISE and INFORMS.

Jianjun Shi received BS and MS degrees in automation from the Beijing Institute of Technology in 1984 and 1987, respectively, and a PhD degree in mechanical engineering from the University of Michigan in 1992. Currently, Dr. Shi is the Carolyn J. Stewart Chair and Professor at the Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology. His research interests include the fusion of advanced statistical and domain knowledge to develop methodologies for modeling, monitoring, diagnosis, and control for complex manufacturing systems. Dr. Shi is a Fellow of the Institute of Industrial and Systems Engineers (IISE), a Fellow of American Society of Mechanical Engineers (ASME), a Fellow of the Institute for Operations Research and the Management Sciences (INFORMS), an elected member of the International Statistics Institute, a life member of the Amreican Astatistics Association (ASA), an Academician of the International Academy for Quality (IAQ), and a member of National Academy of Engineers (NAE).

References

Ahmed, N., Natarajan, T. and Rao, K.R. (1974) Discrete cosine transform. *IEEE Transactions on Computers*, **100**, 90–93.

Apley, D.W. and Shi, J. (1998) Diagnosis of multiple fixture faults in panel assembly. *Journal of Manufacturing Science and Engineering*, **120**, 793–801.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, **3**, 1–122.

Buckley, M. (1994) Fast computation of a discretized thin-plate smoothing spline for image data. *Biometrika*, **81**, 247–258.

De Witte, H., Passefort, S., Besling, W., Maes, J., Eason, K., Young, E., Rittersma, Z. and Heyns, M. (2003) In-line electrical metrology for high-K gate dielectrics deposited by atomic layer CVD. *Journal of the Electrochemical Society*, **150**, F169–F172.

Ding, Y., Ceglarek, D. and Shi, J. (2000) Modeling and diagnosis of multistage manufacturing processes: Part I: State space model, in *Proceedings of the 2000 Japan/USA Symposium on Flexible Automation*, American Society of Mechanical Engineers, New York, pp. 23–26.

Fazel, M., Hindi, H. and Boyd, S.P. (2001) A rank minimization heuristic with application to minimum order system approximation, in *Proceedings of the American Control Conference*, Citeseer, Arlington, VA, pp. 4734–4739.

Gahrooei, M.R., Yan, H., Paynabar, K. and Shi, J. (2020) Multiple tensor-on-tensor regression: an approach for modeling processes with heterogeneous sources of data. *Technometrics*, **62**(1), 1–23.

Huang, C.Y., Chiu, C.F., Wu, W.B., Shih, C.L., Huang, C.C.K., Huang, H., Choi, D., Pierson, B. and Robinson, J.C. (2012) Overlay control methodology comparison: field-by-field and high-order methods. In *Metrology, inspection, and process control for microlithography* XXVI, Vol. 8324, pp. 832427-1–832427-9. International Society for Optics and Photonics. San Jose, California.

Jin, R. and Shi, J. (2012) Reconfigured piecewise linear regression tree for multistage manufacturing process control. *IIE Transactions*, **44**, 249–261.

Lee, T.Y., Lee, B.H., Chin, S.B., Cho, Y.S., Hong, J.S., Hong, J.S. and Song, C.L. (2006) Study of critical dimension and overlay measurement methodology using SEM image analysis for process control, in *Metrology, Inspection, and Process Control for Microlithography XX*, International Society for Optics and Photonics, San Jose, pp. 61522E-1–61522E-8.

Li, J. and Shi, J. (2007) Knowledge discovery from observational data for process control using causal Bayesian networks. *IIE Transactions*, **39**, 681–690.

Li, Y., Sun, H., Deng, X., Zhang, C., Wang, H.-P. and Jin, R. (2020) Manufacturing quality prediction using smooth spatial variable selection estimator with applications in aerosol jet® printed electronics manufacturing. *IISE Transactions*, **52**, 321–333.

Nishi, Y. and Doering, R. (2000) *Handbook of Semiconductor Manufacturing Technology*, CRC Press, Boca Raton, FL.

O'Sullivan, F. (1991) Discretized Laplacian smoothing by Fourier methods. *Journal of the American Statistical Association*, **86**, 634–642.

Parikh, N. and Boyd, S. (2014) Proximal algorithms. *Foundations and Trends® in Optimization*, **1**, 127–239.

Ramsay, J.O. (1988) Monotone regression splines in action. *Statistical Science*, **3**, 425–441.

Ramsay, J.O. (2005) *Functional data analysis*. Springer Series in Statistics, Springer, 2nd ed., New York, p. 426.

Shi, J. (2006) *Stream of Variation Modeling and Analysis for Multistage Manufacturing Processes*, CRC Press, Boca Raton, FL.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B* (*Statistical Methodology*), **67**, 91–108.

Wahba, G. (1990) *Spline Models for Observational Data*, SIAM, Philadelphia, PA.

Yan, H., Paynabar, K. and Shi, J. (2014) Image-based process monitoring using low-rank tensor decomposition. *IEEE Transactions on Automation Science and Engineering*, **12**, 216–227.

Yan, H., Paynabar, K. and Shi, J. (2017) Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics*, **59**, 102–114.

Yuan, M., Ekici, A., Lu, Z. and Monteiro, R. (2007) Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B* (*Statistical Methodology*), **69**, 329–346.

Yuan, M. and Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, **68**, 49–67.

Yue, X., Park, J.G., Liang, Z. and Shi, J. (2020) Tensor mixed effects model with application to nanomanufacturing inspection. *Technometrics*, **62**(1), 116–129.

Zhang, C., Yan, H., Lee, S. and Shi, J. (2018a) Dynamic multivariate functional data modeling via sparse subspace learning. *arXiv preprint arXiv:1804.03797*.

Zhang, C., Yan, H., Lee, S. and Shi, J. (2018b) Multiple profiles sensor-based monitoring and anomaly detection. *Journal of Quality Technology*, **50**, 344–362.