Journal of Quality Technology

A Quarterly Journal of Methods, Applications and Related Topics

ISSN: 0022-4065 (Print) 2575-6230 (Online) Journal homepage: https://www.tandfonline.com/loi/ujqt20

Multiple profiles sensor-based monitoring and anomaly detection

Chen Zhang, Hao Yan, Seungho Lee & Jianjun Shi

To cite this article: Chen Zhang, Hao Yan, Seungho Lee & Jianjun Shi (2018) Multiple profiles sensor-based monitoring and anomaly detection, Journal of Quality Technology, 50:4, 344-362, DOI: 10.1080/00224065.2018.1508275

To link to this article: https://doi.org/10.1080/00224065.2018.1508275

Published online: 31 Oct 2018.

JOURNAL OF QUALITY TECHNOLOGY
2018, VOL. 50, NO. 4, 344–362
https://doi.org/10.1080/00224065.2018.1508275

RESEARCH PAPER

Multiple profiles sensor-based monitoring and anomaly detection

Chen Zhang[a], Hao Yan[b], Seungho Lee[c], and Jianjun Shi[d]

[a]Department of Industrial Engineering, Tsinghua University, Beijing, China; [b]School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, Arizona; [c]Samsung Electronics, Suwon, South Korea; [d]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia

ABSTRACT

Generally, in an advanced manufacturing system hundreds of sensors are deployed to measure key process variables in real time. Thus it is desirable to develop methodologies to use real-time sensor data for on-line system condition monitoring and anomaly detection. However, there are several challenges in developing an effective process monitoring system: (i) data streams generated by multiple sensors are high-dimensional profiles; (ii) sensor signals are affected by noise due to system-inherent variations; (iii) signals of different sensors have cluster-wise features; and (iv) an anomaly may cause only sparse changes of sensor signals. To address these challenges, this article presents a real-time multiple profiles sensor-based process monitoring system, which includes the following modules: (i) preprocessing sensor signals to remove inherent variations and conduct profile alignments, (ii) using multichannel functional principal component analysis (MFPCA)–based methods to extract sensor features by considering cluster-wise between-sensor correlations, and (iii) constructing a monitoring scheme with the top-*R* strategy based on the extracted features, which has scalable detection power for different fault patterns. Finally, we implement and demonstrate the proposed framework using data from a real manufacturing system.

KEYWORDS

data fusion; functional PCA; multichannel profile monitoring; statistical process control

1. Introduction

Advanced automatic data collection and inspection technologies, which generate large amounts of high-dimensional data streams from multiple sensors characterizing the process, have been widely adopted in manufacturing nowadays. Those multiple sensor data provide opportunities for on-line process monitoring and anomaly detection. However, several challenges should be addressed to develop such an on-line monitoring system.

The first and most crucial challenge is how to process high-dimensional, cross-correlated streaming data. These high-dimensional data streams lead to a large number of process parameters to be estimated and consequently require a very large in-control (IC) reference sample size. However, only limited reference samples can be gathered in a manufacturing process. When the data dimension is larger than the reference sample size, we face the curse of dimensionality: it is prohibitive to estimate the joint process distribution, or even its covariance matrix, with limited reference samples. At this point traditional multivariate statistical process control (SPC) methods, such as the Hotelling T^2 chart, will become infeasible, which triggers high demand for new SPC methods.

The second challenge is that multiple variation patterns exist in a manufacturing process. For example, if the process is self-controlled, sensor signals of different products (samples) may be unsynchronized and may have different time lengths due to sample-to-sample variations. Meanwhile, the manufacturing equipment may undergo long-term drifts: sensor signals of the first and consequent products may be at different levels even though all the products are within the prespecified tolerances (i.e., in control). As such, we need to consider how to preprocess raw sensor signals to remove these system-inherent variations for accurate comparison and analysis.

The third challenge is that a complicated correlation structure typically exists among high-dimensional streaming data. First, data produced by each sensor are nonstationary and strongly auto-correlated, which can be regarded as profile (functional) data with within-sensor (within-profile) correlations over time. Since traditional SPC schemes assume that the collected data are independent and identically

CONTACT Jianjun Shi jianjun.shi@isye.gatech.edu Room 109, Groseclose Building, 765 Ferst Drive, Atlanta, GA 30332-0205, USA.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/ujqt.

distributed (*i.i.d.*), these SPC schemes cannot be applied here directly. Second, on the one hand, sensors from the same source may have similar features (i.e., be cross-correlated). Although monitoring each sensor separately could reduce data dimensionality, this method neglects sensor cross-correlations and will eventually sacrifice the detection power. In other words, while separate monitoring might still be useful for detecting change patterns occurring in each single sensor profile, this method is ineffective in detecting changes of sensor cross-correlations. On the other hand, sensor profiles from different sources can show quite different features. Since different data features require particularly tailored analysis methods, different control charts should be designed separately to fit different data features. Therefore, it may be suboptimal if we directly monitor all sensors jointly using one unique control chart without considering data feature difference. With this in mind, we need to strike a balance between separate and joint monitoring of multiple sensor profiles.

The last but not the least challenge is that only a small number of sensor profiles will be influenced or shifted when the manufacturing process goes out of control. Since the influenced sensors or signals are sparse among the high-dimensional process data, the challenges in change detection is increased. Monitoring all sensor profiles without an effective sensor-selection technique may result in detection power losses.

In summary, it is a challenging task to design a unified scheme to monitor multiple sensor profiles. This article will address those challenges and develop a monitoring system. The following subsections will provide a motivation example, a literature review, and a summary of our contributions.

1.1. A motivation example

We use real sensor data from an advanced manufacturing process as an example to illustrate the aforementioned challenges. The manufacturing equipment has 26 sensors, denoted as V1 to V26, that record values of different process variables such as electricity, liquid/gas flow rate, pressure, and so on. Figure 1 shows profiles of six selected sensors over 52 samples from six different product lots. First, it is clear that profile lengths vary from sample to sample, indicating the un-synchronization problem for different samples due to the self-controlled process.

Second, all the profiles have smooth patterns over time, which show their strong within-profile correlations. To better demonstrate this point, we first do data alignment for different samples and then calculate signal correlations of different time points (i.e., within-profile correlations) for every sensor. Figure 2 shows the within-profile correlation structure of V1, V6, and V14. In particular, all these sensors show strong within-profile correlations, and correlations of different sensors are quite different. This indicates that different sensors are driven by different system mechanisms and illustrate different features, which will be explained later in more detail.

Third, besides within-profile correlations, different sensors have complex interprofile relationships. As shown in Figure 1, certain sensors, such as V1 and V26, share similar profile features. This is because these sensors measure some physically related process variables and are located near to each other. However, some other sensors, such as V2 and V12, have profiles with quite different features. This is because that these sensors are located in different places or measure different process variables. The feature similarity can be evaluated to some degree by sensor cross-correlations. Sensors with similar features have strong cross-correlations, while sensors with different features have weak ones. Consequently, as the number of sensors increases the chance that all sensors are strongly correlated becomes quite small, especially for high-dimensional cases. In other words, these multiple (multichannel) profiles can be naturally regarded as multimode data or mixture data. To make this point more explicit, for every sensor we first synchronize profiles of all the samples to the same length using dynamic time warping (Keogh and Ratanamahatana 2005). Then we take the average of the synchronized profiles to get the template profile of every sensor. We subtract the sensor template profile from the synchronized profile to get the corresponding residual profile for each sensor. Finally, we use these residual profiles to calculate cross-correlations of these 26 sensors. From Figure 3 we can see that the cross-correlations have a blocky structure, based on which sensors can be naturally divided into several clusters with strong within-cluster cross-correlations. Furthermore, Figure 4 presents the residual profiles of V1. They have significant time-dependent variance. This is principally caused by fabrication "on-off" operations that lead sensor signals to change drastically at some specific time points where larger fluctuations are expected than at other time points.

Finally, to demonstrate system-inherent variations, Figure 5 shows profiles of V4 over the 52 samples, where sample colors have been arranged to change from light blue (first sample) to purple (last sample)

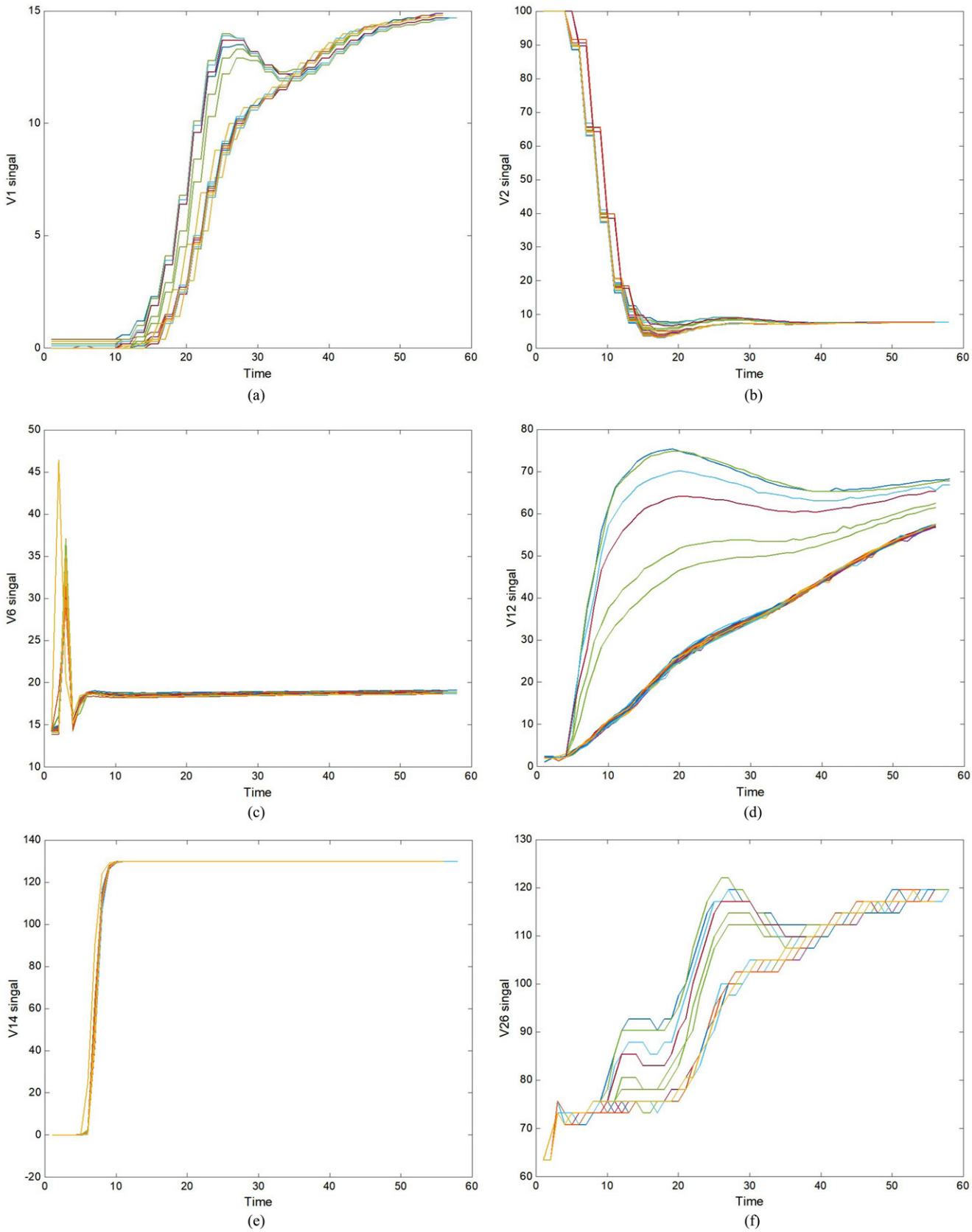

Figure 1. The six selected sensor profiles over 52 samples. (a) V1 (b) V2 (c) V6 (d) V12 (e) V14 (f) V26.

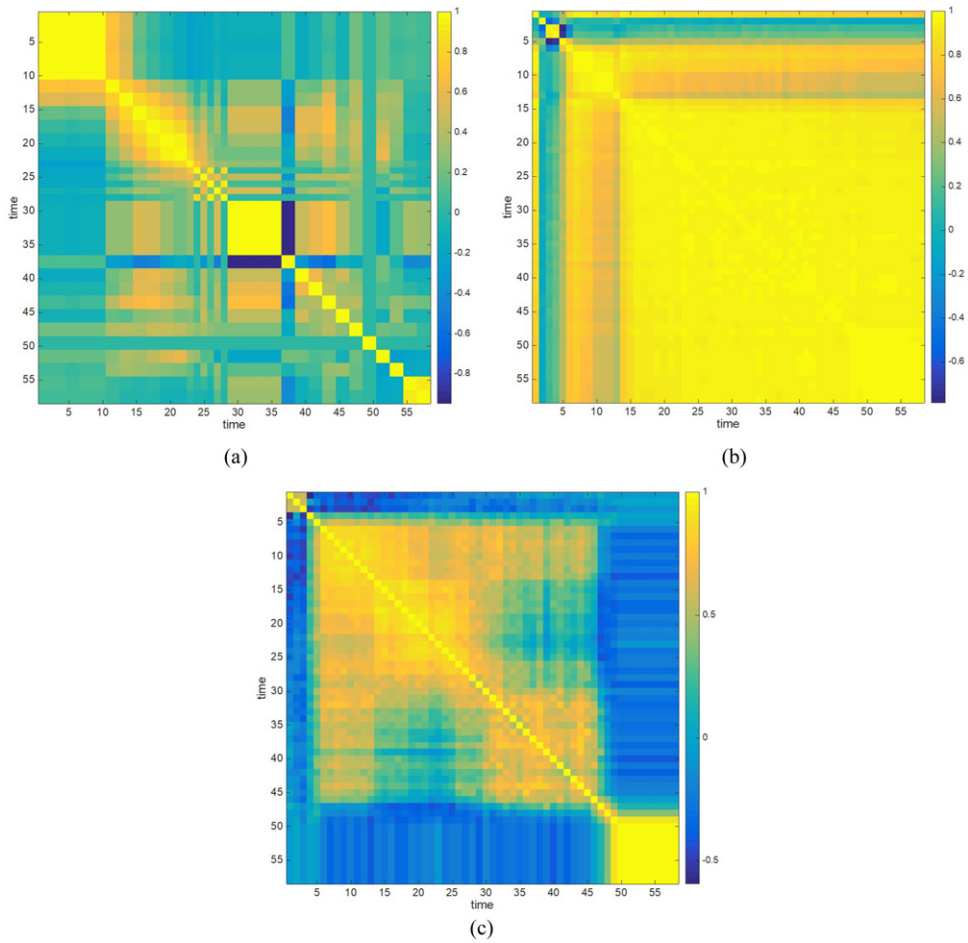

Figure 2. The within-profile correlations of V1, V6, and V14. (a) V1 (b) V6 (c) V14.

according to the actual sample production start time. Obviously there is a long-term drift from the beginning to the end of the production process that shifts the profiles downward. This drift may be caused by the change of equipment status due to the environmental disturbance over time; hence, it should be regarded as normal operations to be taken into account in the monitoring scheme.

1.2. Literature review

In early works of profile data monitoring, Jin and Shi (1999, 2001) proposed to use wavelets to extract features of stamping tonnage signals for on-line fault detection. Since then the topic of profile monitoring has received more and more attention. The current literature can be generally classified into two major categories according to profile shape complexities: linear and nonlinear profile monitoring. For linear profiles, linear regression models are usually used to represent profile data, and regression coefficients are used for monitoring (Mahmoud and Woodall 2004; Zou, Tsung, and Wang 2007). However, these models

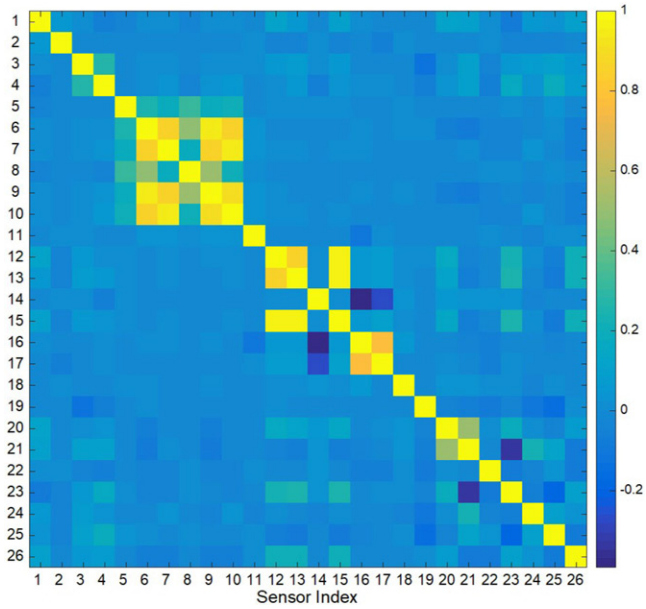

Figure 3. The cross-correlation structure of different sensors.

assume *i.i.d.* measurement noises within a profile. Later, to further effectively account for within-profile correlations, Jensen, Birch, and Woodall (2008)

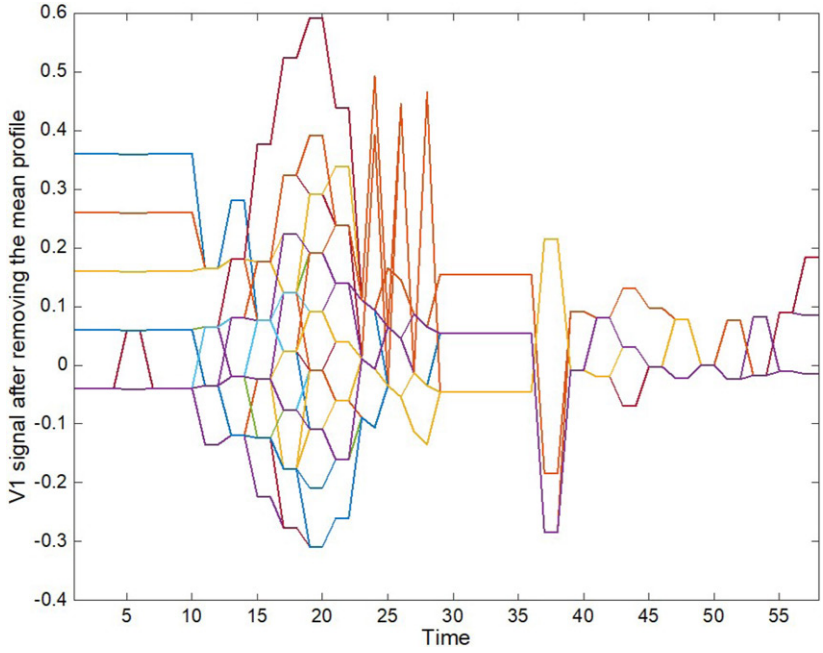

Figure 4. The residual profiles of V1.

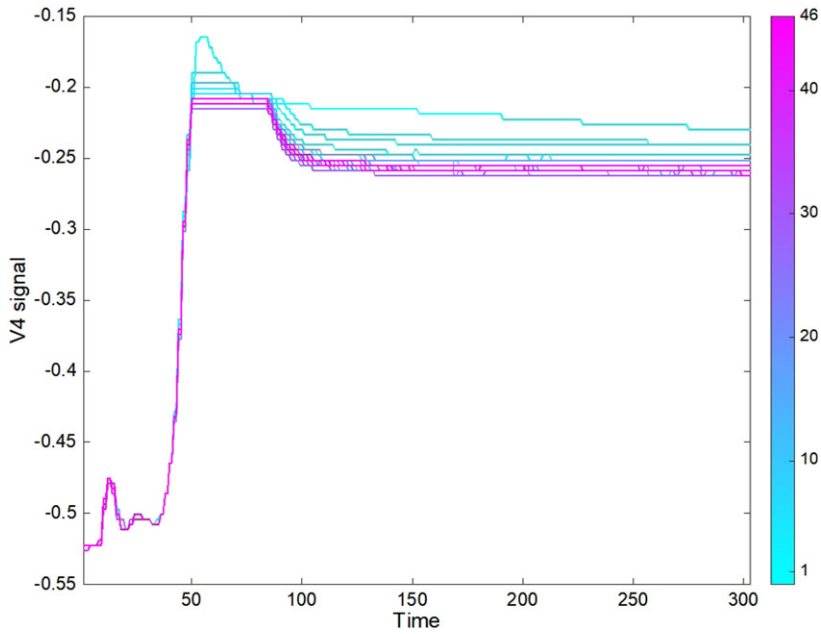

Figure 5. Long-term drift of V4.

proposed a linear mixed-effects model. For nonlinear profiles, nonparametric regression methods have been commonly used in profile monitoring, such as the spline regression (Walker and Wright 2002), the wavelet regression (Zhou, Sun, and Shi 2006), and the local linear smoothers (Zou, Tsung, and Wang 2008). Qiu, Zou, and Wang (2010) and Paynabar and Jin (2011) introduced nonlinear mixed-effects models that also considered within-profile correlations. Beyond these models, Ding, Zeng, and Zhou (2006) considered dimension reduction and data-clustering techniques for Phase I monitoring of nonlinear profiles. For more detailed discussions about related methods, please refer to Noorossana, Saghaei, and Amiri (2011) and Woodall (2007) for a comprehensive review. However, the literature mentioned above focuses only on process with a univariate profile. Though for multichannel profiles we may apply these methods to every channel separately, this completely ignores cross-correlations of multiple channels and weakens the detection power.

In the literature, few studies have been conducted on monitoring multiple profiles simultaneously. For linear profiles with explanatory variables, Noorossana et al. (Noorossana, Eyvazian, Vaghefi 2010; Noorossana, Eyvazian, Amiri et al. 2010) and Ghashghaei and Amiri (2016) proposed several monitoring schemes based on the ordinary least square method. Zou, Ning, and Tsung (2012) proposed a LASSO-based monitoring scheme. For nonlinear profiles, and since signals are usually high-dimensional, dimension-reduction methods are normally applied first to reduce data dimensionality. For example, Kim et al. (2006) introduced a multichannel profile monitoring scheme using principal curves. Paynabar, Jin, and Pacella (2013) proposed to use uncorrelated multilinear principal component analysis (UMPCA) for feature extraction and fault detection. Later, Grasso, Colosimo, and Pacella (2014) compared multiple PCA-based methods, such as multilinear PCA (MPCA) and vectorized PCA (or multiway PCA, shortened as VPCA), for monitoring. Recently, Paynabar et al. (2016) further applied functional data analysis to profile monitoring and constructed a change-point model based on multichannel functional PCA (MFPCA). However, all these aforementioned methods typically assume that multichannel profiles have strong cross-correlations with similar features and cannot effectively handle profiles with different features. This is because their extracted PCA loadings would mix different features together. Consequently, these methods' monitoring performances will deteriorate severely (as demonstrated in our case study). To our best knowledge, so far no monitoring scheme has been developed for profiles with different features from either a statistical or practical perspective.

1.3. Our contributions

Motivated by the previous work, the goal of this research is to design a systematic framework for online monitoring of high-dimensional streaming data in a manufacturing system. Our contributions are summarized here: (1) We propose a preprocessing framework for manufacturing data considering their characteristics. In particular, for the product fabrication time variation problem, unless directly using naive interpolation methods (Lee et al. 2011), which brutally break profile patterns for different samples, we propose using dynamic time warping to ensure the patterns are maximally preserved and the additional noise is minimally introduced. For long-term drift removal, we propose a fixed-effects model to automatically learn drifting sensors, drifting time, and drifting magnitudes. (2) After preprocessing, we propose a sensor clustering algorithm according to the sensor cross-correlation structure. Based on this algorithm, sensors within each cluster have strong correlations, while sensors between clusters have no (or weak) correlations. In this way we gather sensors with similar features together as a cluster and get several clusters with different features. (3) Since there is no more within-cluster feature difference, we can monitor sensor data within a cluster together and monitor each cluster respectively. In particular, MFPCA is applied to sensors of each separate cluster to extract their shared features. Then the MFPCA scores together with the residuals are used to construct local monitoring statistics to detect local changes in this cluster. (4) Finally, we propose a data-fusion strategy to incorporate these local monitoring results together as the final monitoring statistic. Specifically, this statistic is based on the top-R rule (i.e., the sum of the largest R local monitoring statistics (Mei 2011)). This top-R rule can not only filter out in-control noise and increase detection sensitivity but also allow for scalable detection power for different change patterns and can assist the diagnostic procedure after an abnormal signal is triggered. The overview of the procedure is shown in Figure 6.

The remainder of this article is organized as follows: Section 2 presents the data-preprocessing methods; Section 3 presents the monitoring scheme based on sensor clustering and data fusion; Section 4 evaluates the charting performance using some numerical studies in a manufacturing process; finally, Section 5 concludes this article with some discussions about future work.

2. Data preprocessing

In this section we will discuss data preprocessing to address two issues: data alignments of multiple profiles and long-term drift removal. Here we denote $i = 1, \ldots, N$; $j = 1, \ldots, P$; and $t = 1, \ldots, T_i$ for the indices of different samples, sensors, and time points, respectively. For example, Y_{ijt} indicates the scalar value of the jth sensor at the tth time point of the ith sample. It should be noted that since the raw profiles of different samples may be unsynchronized, T_i are different for different samples.

2.1. Data synchronization

As mentioned in Section 1, profile data has unsynchronized different lengths for different samples

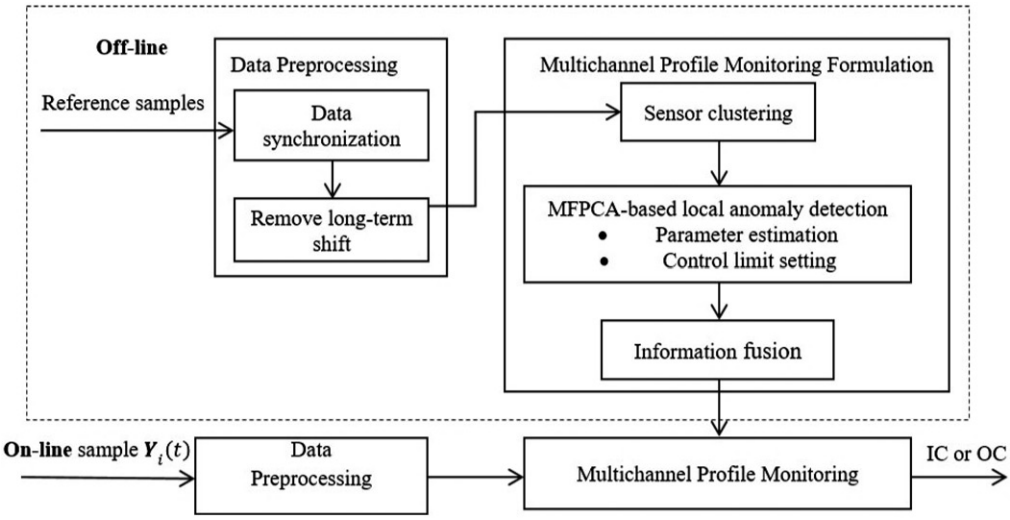

Figure 6. An overview of the developed monitoring system.

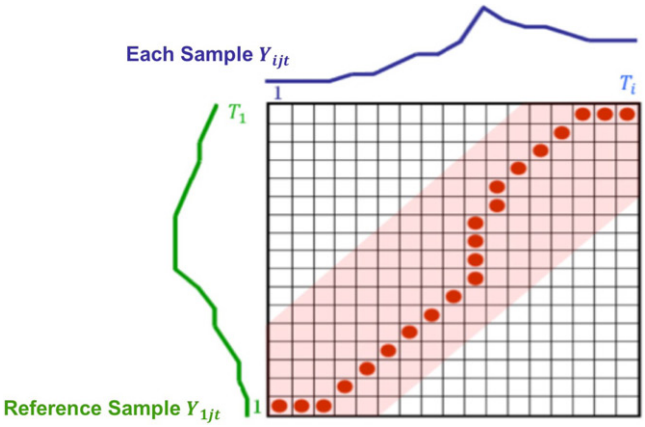

Figure 7. Dynamic time-warping procedure.

due to the self-controlled process in the manufacturing process. In this section, for every sensor j we would like to synchronize the time stamp of the ith sample by comparing its signals $Y_{ijt},\ t = 1, ..., T_i$ with the reference sample denoted by $Y_{1jt},\ t = 1, ..., T_1$ (here we set the first sample as the reference) using dynamic time warping (DTW; Keogh and Ratanamahatana 2005). To align these two sequences using DTW, we first construct a T_1-by-T_i distance matrix where the (t_1, t_i) element corresponds to the square distance: $d(t_1, t_i) = (Y_{1jt_1} - Y_{ijt_i})^2$. DTW aims to solve the optimal path $\left(t_1^l, t_i^l\right)_{l=1,...,L}$ that minimizes the warping cost:

$$DTW(\boldsymbol{Y}_{1j}, \boldsymbol{Y}_{ij}) = \min\left\{\sum_{l=1}^{L} d\left(t_1^l, t_i^l\right)\right\},$$

where (t_1^l, t_i^l) is the index of the lth element on the alignment path for the ith sample.

Furthermore, we constrain the search space by the Sakoe-Chiba band constraint, that is,

$$\left|t_1^l - t_i^l\right| \le \max(0.2T_1, 0.2T_i, |T_1 - T_i|), l = 1, ..., L,$$

such that the optimal path is not too far from the diagonal line, as shown in Figure 7. The optimal path can then be found efficiently by dynamic programming. Finally, the optimal path provides a mapping from the ith sample to the reference sample such that $(t_1^l, t_i^l)_{l=1,...,L}$ is the optimal path that achieves the minimal warping cost. Therefore, profiles of each sample can be aligned with those of the reference sample. In Figure 8 we show one example of how DTW can synchronize profiles of different samples.

2.2. Removing long-term drift

As shown in Figure 5, long-term drift magnitude varies in different time segments even for the same sensor. This phenomenon motivates us to estimate long-term drifts for different time segments separately. Generally, the long-term drift depends on the system status, which is influenced by the "on-off" operations. With this in mind, we may treat the time points between two sequential "on-off" operations as a nature segment. Specifically, after synchronization all samples share the same time length T. These T time points can be separated into M steps, with Y_{ijt} given by

$$Y_{ijt} = \mu_{jt} + b_{ijm} + \epsilon_{ijt},\ m = 1, ...,\ M, \qquad [1]$$

where m denotes the corresponding segment to which the tth time point belongs. We define t_m as the last time point of the mth segment (i.e., for the mth segment of the profile, $t_{m-1} < t \le t_m,\ m = 1, ...,\ M$ with $t_0 = 0$). In Eq. [1], μ_{jt} is the template signal of the jth sensor at the tth time point. b_{ijm} is assumed to be the constant drift of the jth variable in the mth time

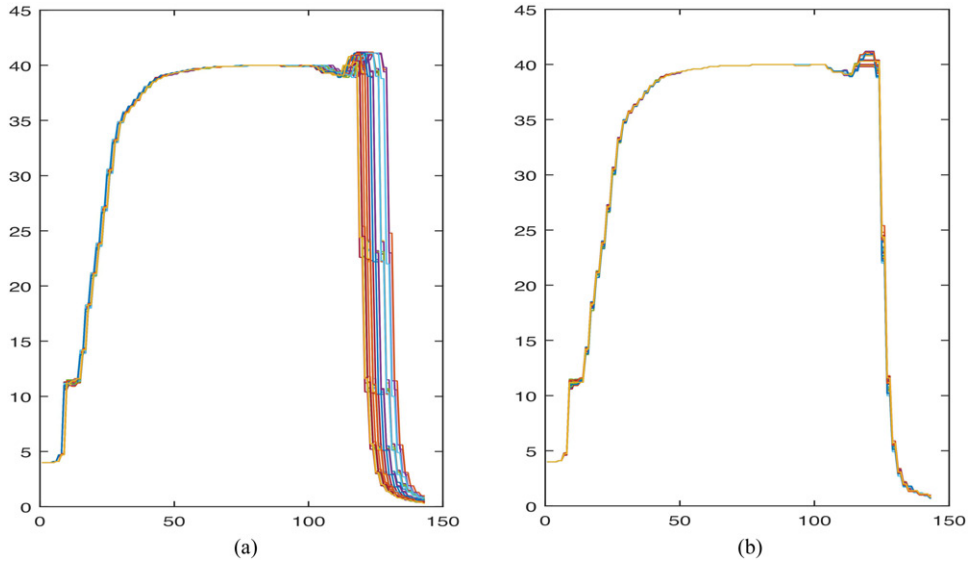

Figure 8. Effect of alignment by DTW on V2. (a) Sample profiles before alignment (b) Sample profiles after alignment.

segment of the ith sample, with respect to the template signals. ϵ_{ijt} denotes the remaining individual signal for $t = 1, \ldots, T$, following a distribution with mean zero and variance σ^2_{jt}. Following Lee et al. (2011), we apply the weighted least squares (WLS) method to estimate parameters μ_{jt} and b_{ijm} by minimizing the weighted sum of squared errors (WSSE), that is,

$$\min_{\mu_{jt}, b_{ijm}} WSSE_{jm} = \sum_{i=1}^{N} \sum_{t=t_{m-1}+1}^{t_m} \frac{1}{\sigma^2_{jt}} \left(Y_{ijt} - \mu_{jt} - b_{ijm}\right)^2, \quad \text{s.t.} \ \sum_{i=1}^{N} b_{ijm} = 0, \tag{2}$$

for $j = 1, \ldots P$, $m = 1, \ldots, M$, separately. Eq. [2] can be solved in a close-form following the same approach in Lee et al. (2011). However, one limitation of this approach in Lee et al. (2011) is that the estimated $\hat{b}_{ijm}$ is not necessarily the long-term drift because it only denotes the constant difference of the ith sample from the template profile in the mth segment caused by sample-to-sample variations. Meanwhile, the true long-term drift $\hat{b}_{ijm}$ should be a drift and expected to be monotone. In other words, the cumulative drift magnitude between neighbor samples, $\left|\sum_{i=2}^{N} (\hat{b}_{ijm} - \hat{b}_{(i-1)jm})\right| = \left|\hat{b}_{Njm} - \hat{b}_{1jm}\right|$, should be large enough when the true long-term drift exists. Therefore, we propose to apply hard thresholding to the estimated $\hat{b}_{ijm}$, as

$$\tilde{b}_{ijm} = \begin{cases} \hat{b}_{ijm}, & if \ \left|\hat{b}_{Njm} - \hat{b}_{1jm}\right| > s_{jm} \\ 0, & otherwise \end{cases}, \tag{3}$$

where s_{jm} is the threshold related to the variance of $\hat{b}_{ijm}$. In our study, we find that $s_{jm} = 3 \times \text{std}\ (\hat{b}_{ijm})$ works very well in practice.

When σ^2_{jt} is unknown, we can estimate it by $\hat{\sigma}^2_{jt} = \sum_{i=1}^{N} \frac{\left(Y_{ijt} - \hat{\mu}_{jt} - \tilde{b}_{ijm}\right)^2}{N}$. Furthermore, it should be noted that when "on-off" operations do not change the profile too much we may combine several sequential segments as one for analysis. From the data-driven perspective, we may redefine the segments by setting $\{t_m | \hat{\sigma}^2_{jt_m} > s\}$ as segment boundaries, where s is a pre-specified threshold. In this article we set $s = 3 \times \text{median}(\hat{\sigma}^2_{jt})$ and then iterate the above procedure to estimate $\hat{b}_{ijm}$, $\tilde{b}_{ijm}$, $\hat{\sigma}^2_{jt}$, *and* t_m. It should be noted that M can change in different iteration steps. This estimation algorithm converges empirically in our numerical studies.

We demonstrate the efficiency of this algorithm using V2 as an example. Figure 9(a) shows the profiles of V2 in one processing step before removing the long-term drift, from which we can clearly see that these profiles have a long-term mean drift because $t = 50$. After iterating the algorithm until convergence, four segment boundaries satisfying $\{t_m | \hat{\sigma}^2_{jt_m} > s\}$ are identified as $t_1 = 50$, $t_2 = 51$, $t_3 = 52$, and $t_4 = 86$ (shown by the vertical dashed lines in Figure 9(a)). However, only the last two segments $[52, 86]$ and $[86, 304]$ have nonzero $\tilde{b}_{ijm}$ to be removed, as shown in Figure 9(b). The final sample profiles after removing the drifts $\tilde{b}_{ijm}$ are shown in Figure 9(c).

3. Multichannel profile monitoring formulation

In this section, we will introduce the multichannel profile monitoring scheme in detail. We treat a profile as a function and adopt functional data analysis

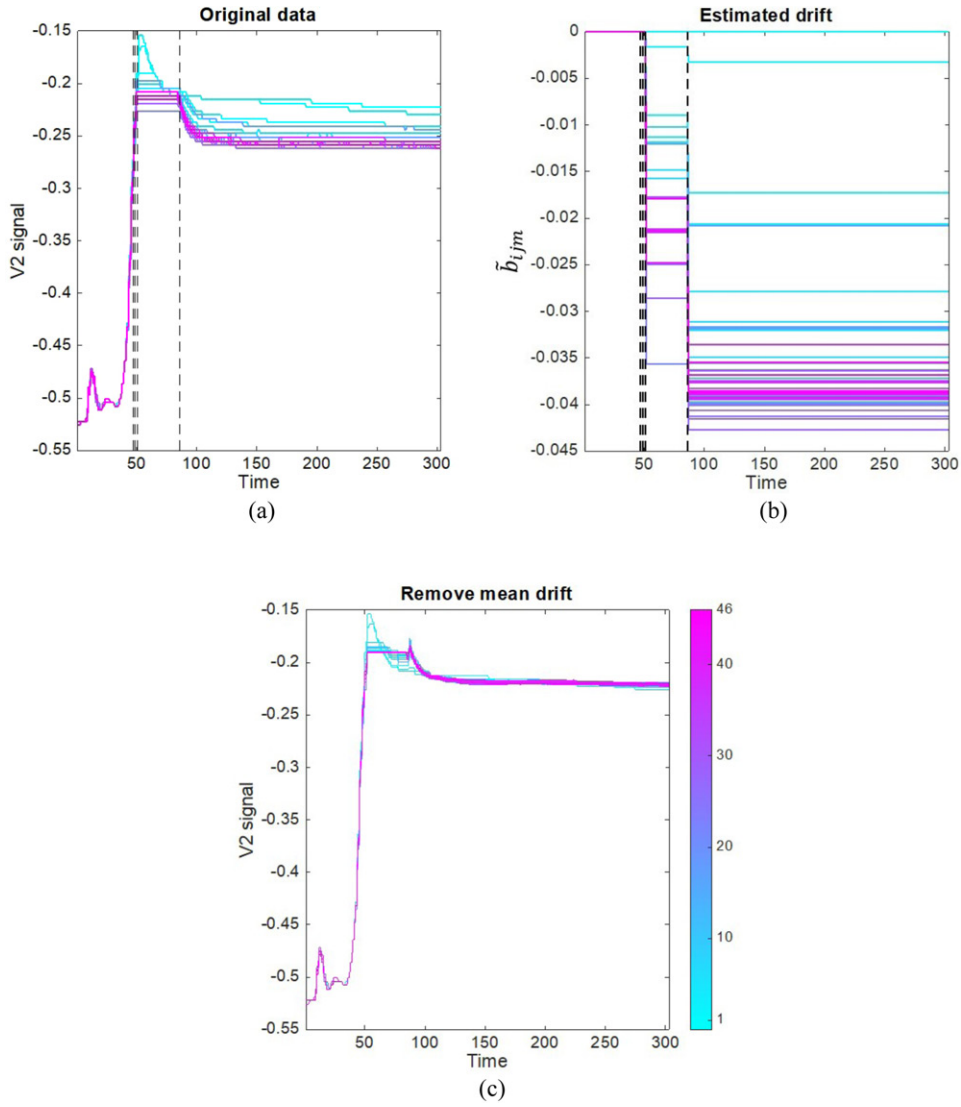

Figure 9. Remove the long-term drift of V2. (a) Original profiles (b) Estimated drift (c) Profiles after removing the drift.

(FDA) for data description and monitoring. In particular, we consider the aligned profile data of the ith sample of the jth sensor as a continuous function $Y_{ij}(t)$ with respect to $t \in \mathcal{T} = [a, b]$, where $[a, b]$ is the manufacturing time interval. The observations Y_{ijt}, $t = 1, ..., T$ are discrete sensing observations of the function at T sampling time points $t = 1, ..., T$. According to Eq. [1], after removing the long-term drift $Y_{ij}(t)$ can be formulated as

$$Y_{ij}(t) = \mu_j(t) + \epsilon_{ij}(t), \quad i = 1, ...N; \ j = 1, ..., P, \qquad [4]$$

where $\mu_j(t)$ is the function of the template profile of the jth sensor solved in Eq. [2] and $\epsilon_{ij}(t)$ is the stochastic error with $E(\epsilon_{ij}(t)) = 0$. Then, profile data of all the sensors can be denoted as $\boldsymbol{Y}_i(t) = [Y_{i1}(t), ..., Y_{iP}(t)]$, with $\boldsymbol{\mu}(t) = [\mu_1(t), ..., \mu_P(t)]$ and $\boldsymbol{\varepsilon}_{,i}(t) = [\epsilon_{i1}(t), ..., \epsilon_{iP}(t)]$. Our goal is to sequentially monitor distributional changes of $\boldsymbol{\mu}(t)$.

When the process is in-control, assume that there are m_0 *i.i.d.* reference samples $\boldsymbol{Y}_{-m_0+1}(t), ..., \boldsymbol{Y}_0(t)$ (or IC samples interchangeably). Then, in the sequential monitoring stage, the subsequent ith on-line testing sample after preprocessing (i.e., $\boldsymbol{Y}_i(t)$) is assumed to follow a change-point model

$$\boldsymbol{Y}_i(t) = \begin{cases} \boldsymbol{\mu}_0(t) + \epsilon_i(t), & for \ i = 1, ..., \tau, \\ \boldsymbol{\mu}_1(t) + \epsilon_i(t), & for \ i = \tau + 1, ..., \end{cases} \qquad [5]$$

where τ is the unknown change point and $\boldsymbol{\mu}_0(t)$ and $\boldsymbol{\mu}_1(t)$ are the IC and OC mean (template) functions. We aim to detect the OC situation as soon as possible and estimate the change point τ as accurate as possible.

As discussed earlier, sensor profiles from the same source have similar features and are cross-correlated. Thus, it is intuitive to consider their correlation structure and use multivariate SPC schemes for joint monitoring. However, sensor profiles from different sources have different features and can be regarded as different clusters. Therefore, it is not effective to construct a universal monitoring scheme without considering feature difference. Furthermore, due to the sparsity of OC variables, jointly monitoring all sensors without selecting correlated clusters may dilute detection power. As Figure 3 shows, the 26 sensors are sparsely correlated with a clusterwise correlation structure. This clusterwise structure naturally motivates us toward the idea of divide and conquer. With this in mind, we propose a monitoring scheme based on sensor clustering and fusion. Its key idea is to first partition multiple sensors into several sensor clusters so that sensors in the same cluster share similar features and feature difference only exists in different clusters. Then we construct local monitoring statistics for each sensor cluster separately by taking advantage of the local correlation structure. Finally, these local monitoring statistics are fused together by summing up the largest R local test statistic for final decision-making. In this way, the united monitoring scheme can, on the one hand, take into account both feature similarity and feature difference. On the other hand, it can filter out IC noise without scarifying detection power too much and therefore improve detection power for sparse changes. Furthermore, it is convenient for us to identify the set of potentially changed sensors by selecting sensors whose groups are included in the largest R local test statistic. This divide and conquer idea has been successfully used in some other SPC schemes (Zhang, Chen, and Zou 2016).

3.1. Sensor clustering

In this subsection we will discuss how to cluster multichannel profiles based on m_0 reference samples $\boldsymbol{Y}_{-m_0+1}(t), \ldots, \boldsymbol{Y}_0(t)$. As mentioned in Section 1.1, sensor profiles can be naturally clustered according to their cross-correlation matrix. Therefore, here we adopt the agglomerative hierarchical correlation clustering method, which begins with treating each sensor as a separate cluster and then successively merging them into larger clusters according to sensor correlations. In each step of hierarchical clustering the algorithm finds the closest pair of clusters and then merges them into a new parent cluster. This is repeated until only one cluster is left after $P-1$ iterations, where P is the number of sensors. Here the Pearson's correlation is used to measure the similarity between different sensors, which can be estimated as

$$\rho_{jj'} = \frac{\sum_{i=-m_0+1}^{0}\sum_{t=1}^{T}\left(Y_{ijt}-\overline{Y}_{jt}\right)\left(Y_{ij't}-\overline{Y}_{j't}\right)}{\sqrt{\sum_{i=-m_0+1}^{0}\sum_{t=1}^{T}\left(Y_{ijt}-\overline{Y}_{jt}\right)^2}\sqrt{\sum_{i=-m_0+1}^{0}\sum_{t=1}^{T}\left(Y_{ij't}-\overline{Y}_{j't}\right)^2}}, \quad [6]$$

for $j, j' = 1, \ldots, P$, where $\overline{Y}_{jt} = \frac{1}{m_0}\sum_{i=-m_0+1}^{0} Y_{ijt}$. Then the distance (dissimilarity) between different sensors is defined as

$$d_{jj'} = 1-|\rho_{jj'}|, \quad j, j' = 1, \ldots, P. \quad [7]$$

We can see that the more correlated the two sensors are, the shorter their distance will be. In addition to the distance between two sensors, cophenetic distance with the average linkage (i.e., $D(r, k) = \frac{\sum_{v\in r}\sum_{u\in k} d_{vu}}{N_r N_k}$) is used to measure the between-cluster distance of clusters r and k, where N_r and N_k are the number of sensors in clusters r and k. Then in every step, according to $D(r, k)$, we can select the two most similar clusters and merge them into one. This sequential merging procedure can be drawn as a dendrogram, based on which we can finally get the sensor information of every cluster given a prespecified number of clusters, G.

The optimal choice of G will balance model accuracy and complexity. In general, a small G leads to larger clusters, less estimation accuracy, and low complexity. The reverse is also true. In real applications, the choice of G can depend on some prior domain knowledge. For example, in our case study the optimal $G = 16$ is selected based on the domain knowledge that the minimum correlation between two sensors in the same cluster is no smaller than 0.2. In general, when no G is preferred in advance some clustering diagnostic methods, such as the elbow method of Thorndike (1953) or the GAP statistic of Tibshirani, Walther, and Hastie (2001) could be used to choose G.

3.2. Multichannel functional PCA–based local anomaly detection

After sensor clustering, assume that we have G clusters in total with each cluster having p^g sensors, where $g = 1, \ldots, G$. Then the profile data for each cluster can be denoted as $\boldsymbol{Y}_i^g(t) = \left[Y_{i1}(t), \ldots, Y_{ip^g}(t)\right]$, with the corresponding template profile $\boldsymbol{\mu}^g(t) = \left[\mu_1(t), \ldots,\right.$

$\mu_{p^g}(t)]$ and stochastic error $\epsilon_i^g(t) = [\epsilon_{i1}(t), \ \ldots, \epsilon_{ip^g}(t)]$, $i = 1, \ldots, N$. Then we discuss how to construct local monitoring statistics for $\boldsymbol{Y}_i^g(t)$ to detect changes of $\boldsymbol{\mu}^g(t)$, $g = 1, \ldots, G$ separately. In the next subsection we will address how to fuse the G local statistics together as the final monitoring statistic to detect changes of $\boldsymbol{\mu}(t)$.

A straightforward monitoring statistic for $\boldsymbol{Y}_i^g(t)$ is to apply the traditional parametric or nonparametric SPC methods. However, in many situations the total number of grid time points (i.e., $T \times p^g$) is much larger than the number of IC reference samples, m_0. We thus suffer from the curse of dimensionality. Hence it is usually important to perform dimension reduction before the monitoring step. One fundamental technique is to use functional PCA (FPCA) to extract a few major and typical features from the functional data. FPCA has been applied in univariate Phase I profile monitoring in Yu, Zou, and Wang (2012). Recently, Paynabar et al. (2016) extended FPCA to multichannel profiles as MFPCA by appropriately addressing their cross-correlations.

In particular, Paynabar et al. (2016) assumed that $\boldsymbol{Y}_i^g(t)$ can be represented by a set of orthonormal Eigen functions as follows,

$$\boldsymbol{Y}_i^g(t) = \boldsymbol{\mu}_0^g(t) + \sum\nolimits_{k=1}^{\infty} \xi_{ik}^g v_k^g(t), \qquad [8]$$

with an explicit form $\xi_{ik}^g = \int_a^b (\boldsymbol{Y}_i^g(t) - \boldsymbol{\mu}_0^g(t)) v_k^g(t) dt$, where $v_k^g(\cdot)$, $k = 1, 2, \ldots \infty$ are the Eigen functions of the covariance function of $\boldsymbol{Y}_i^g(t)$ (i.e., $c^g(t,s) = E\left[(\boldsymbol{Y}^g(t) - \boldsymbol{\mu}_0^g(t))(\boldsymbol{Y}^g(s) - \boldsymbol{\mu}_0^g(s))\right]$) with the corresponding Eigen values $\lambda_k^g, k = 1, 2, \ldots \infty$. In particular, $c^g(t,s)$ can be estimated using the m_0 reference samples as

$$\hat{c}^g(t,s) = \frac{1}{m_0} \sum\nolimits_{i=-m_0+1}^{0} \sum\nolimits_{j=1}^{p^g} \left(Y_{ij}^g(t) - \mu_{0j}^g(t)\right)\left(Y_{ij}^g(s) - \mu_{0j}^g(s)\right). \qquad [9]$$

When $\mu_{0j}^g(t)$ is unknown in practice, we substitute it with its estimate $\hat{\mu}_{0j}^g(t) = \sum_{i=-m_0+1}^{0} Y_{ij}^g(t)/m_0$ using the historical reference data. Then the corresponding estimators of $v_k^g(\cdot)$ & λ_k^g are defined as

$$\int_a^b \hat{c}^g(t,s)\ \hat{v}_k^g(s) ds = \hat{\lambda}_k^g \hat{v}_k^g\ (t), \ \ t \in \mathcal{T}, \ \ k = \ 1, 2, \ldots, \infty. \qquad [10]$$

Under some mild conditions, $\hat{c}^g(t,s)$, $\hat{v}_k^g(t)$, and $\hat{\lambda}_k^g$ are consistent estimators of $c^g(t,s)$, $v_k^g(t)$, and λ_k^g (Paynabar et al. 2016).

The form of Eq. [8] indicates that all the p^g profiles share a common set of Eigen functions, and their cross-correlations are essentially described by the correlations of ξ_{ik}^g. In particular, we assume $\boldsymbol{\xi}_{ik}^g \in \mathbb{R}^{p^g}$ follows a p^g-dimensional normal distribution with mean zero and covariance matrix $\boldsymbol{\Sigma}_k^g$, which can be estimated using the m_0 reference samples as

$$\hat{\boldsymbol{\Sigma}}_k^g = \frac{1}{m_0} \sum\nolimits_{i=-m_0+1}^{0} \int_a^b \left(\boldsymbol{Y}_i^g(t) - \boldsymbol{\mu}_0^g(t)\right) \hat{v}_k^g\ (t) dt \times \int_a^b \left(\boldsymbol{Y}_i^g(t) - \boldsymbol{\mu}_0^g(t)\right)^T\ \hat{v}_k^g(t) dt. \qquad [11]$$

Furthermore, since $\hat{\boldsymbol{\xi}}_{ik}^g$ is the projection of $(\boldsymbol{Y}_i^g(t) - \boldsymbol{\mu}_0^g(t))$ to the kth Eigen function, it is an ideal indicator that reflects the difference between the ith sample $\boldsymbol{Y}_i^g(t)$ and $\boldsymbol{\mu}_0^g(t)$. In particular, $\hat{\xi}_{ik}^g$ shows the amount of deviation of the ith sample from the kth variation mode. If $\boldsymbol{Y}_i^g(t)$ is out of control, $\hat{\boldsymbol{\xi}}_{ik}^g \to^d N\left(\int_a^b \left(\boldsymbol{\mu}_1^g(t) - \boldsymbol{\mu}_0^g(t)\right)\ \hat{v}_k^g(t) dt, \boldsymbol{\Sigma}_k^g\right)$ (Paynabar et al. 2016). In this way we can construct a hypothesis test using $\hat{\boldsymbol{\xi}}_{ik}^g$. Here we choose the largest ξ, $k = 1, \ldots, d$ for monitoring. To further improve the detection performance on small mean shifts, we adopt the exponential weighted moving average (EWMA) strategy and define

$$\boldsymbol{X}_i^g(t) = (1-w)\boldsymbol{X}_{i-1}^g(t) + w\boldsymbol{Y}_i^g(t), \qquad (12)$$

$$\boldsymbol{z}_{ik}^g = \int_a^b \left(\boldsymbol{X}_i^g(t) - \boldsymbol{\mu}_0^g(t)\right) \hat{v}_k^g\ (t) dt, \ \ k = 1, \ldots, d, \qquad [13]$$

where $\boldsymbol{X}_0^g(t) = 0$ and w is the exponential weighting parameter. In practice, the common choice of w is in the interval $[0.05, 0.2]$. Thus, the final test statistic for the ith on-line sample is defined as

$$Z_i^g = \sum\nolimits_{k=1}^{d} \boldsymbol{z}_{ik}^{gT} \left(\hat{\boldsymbol{\Sigma}}_k^g\right)^{-1} \boldsymbol{z}_{ik}^g. \qquad [14]$$

It should be noted that Z_i^g can detect only changes that are captured by $\hat{v}_k^g\ (t)$, $k = 1, \ldots, d$. For changes that are captured by higher-order projections such as $\hat{v}_k^g\ (t)$, $k = d+1, \ldots,$ we propose a corresponding residual chart for Z_i^g similar to the residual chart of traditional PCA-based monitoring schemes (Jackson and Mudholkar 1979). Specifically, we calculate the residual of the approximation of $\boldsymbol{Y}_i^g(t)$ using the first d Eigen functions

$$\boldsymbol{e}_i^g(t) = \ \boldsymbol{Y}_i^g(t) - \sum\nolimits_{k=1}^{d} \hat{\boldsymbol{\xi}}_{ik}^g \hat{v}_k^g\ (t), \qquad [15]$$

and its EWMA form as

$$\boldsymbol{r}_i^g(t) = (1-w)\boldsymbol{r}_{i-1}^g(t) + w\boldsymbol{e}_i^g(t), \qquad [16]$$

with $\boldsymbol{r}_0^g(t) = 0$. Then we construct the residual chart for the ith sample using the square of the norm of

$\boldsymbol{r}_i^g(t)$ as

$$Q_i^g = \sum_{j=1}^{p^g} ||\boldsymbol{r}_{ij}^g||^2 = \sum_{j=1}^{p^g} \int_a^b \left\{ r_{ij}^g(t) \right\}^2 dt. \quad [17]$$

For a cluster with a single sensor (i.e., $p^g = 1$), MFPCA degenerates to the traditional FPCA. Then the procedure above (Eqs. [8]–[17]) is a Phase II control chart for univariate profile based on FPCA, which is the counterpart of the Phase I control chart of Yu, Zou, and Wang (2012).

3.3. Information fusion via top-R thresholding rule

Suppose that we have divided all the P sensors into G clusters and calculated $\{Z_i^g, Q_i^g\}$, $g = 1, ..., G$. The next problem is how to combine these local monitoring statistics for each cluster to produce a global monitoring statistic to detect OC conditions as quickly as possible, subject to a prespecified systemwise false alarm rate. An intuitive way to construct the global monitoring statistic is to sum all local monitoring statistics together (Mei 2010). However, though this global monitoring statistic usually performs well for small mean shifts among all the sensors, it is not suitable for the case of mean shifts that occur only in a few sensors, due to the additional noise introduced by the local monitoring statistics for unchanged sensors. As a complementary therapy, the maximum chart (Tartakovsky et al. 2006) aims to detect large but sparse shifts by selecting only the largest local monitoring statistic. However, this selection clearly displays its poor performance if more than one sensor changes. Mei (2011) and Liu, Mei, and Shi (2015) proposed a top-R rule to balance between the sum and the maximum chart by summing the largest R local monitoring statistics as the global monitoring statistic. Furthermore, this top-R rule is particularly effective when we have some prior domain knowledge that, at most, R out of G data streams will be affected by an abnormal event. Motivated by this, we propose to fuse the G local monitoring statistics with the top-R rule. However, since p^g are different for different clusters, we need to normalize these local monitoring statistics first as $\{\tilde{Z}_i^g, \tilde{Q}_i^g\}$, $g = 1, ..., G$, to make them share the same mean zero and standard deviation one, that is,

$$\tilde{Z}_i^g = \frac{Z_i^g - E(Z_i^g)}{std(Z_i^g)}, \quad \tilde{Q}_i^g = \frac{Q_i^g - E(Q_i^g)}{std(Q_i^g)}. \quad [18]$$

Then we rank the test statistics $\tilde{Z}_i^{(1)} \geq ... \geq \tilde{Z}_i^{(G)}$, $\tilde{Q}_i^{(1)} \geq ... \geq \tilde{Q}_i^{(G)}$ and propose to fuse these monitoring statistics as

$$T_i = \sum_{r=1}^{R} \tilde{Z}_i^{(r)}, \quad W_i = \sum_{r=1}^{R} \tilde{Q}_i^{(r)}. \quad [19]$$

Then, if $T_i > h_T$ *or* $W_i > h_W$ the monitoring scheme triggers an OC alarm for the ith sample. Here h_T and h_W are the control limits according to the prespecified systemwise IC average run length (ARL_0). In particular, we set h_T and h_W by simulation to ensure that the separate T chart and W chart have the same *IC ARL*, which is almost equal to $2 \times ARL_0$; in the meantime, the joint T and W chart has the *IC ARL* equal to ARL_0.

In general, the choice of R depends on the specific OC scenario of most interest. As mentioned in Mei (2011) and Liu, Mei, and Shi (2015), a larger R leads to a better detection performance for global anomaly patterns that may occur in a lot of sensors but with small magnitudes, while a smaller R results in a better detection performance for extreme anomaly patterns that occur in only few sensors but with large magnitudes.

4. Case study

In this section we revisit the manufacturing process in Section 1 and use it as a case study to illustrate the proposed monitoring framework. As mentioned earlier, $P = 26$ sensors of different types, denoted as V1 to V26, are used to monitor the process. There are 46 IC samples in the data set. We first applied DTW to remove the nonsynchronization effect for different samples; after, the samples have the time length $T = 58$. Then we removed the long-term drift using the data-driven approach proposed in Section 2.2.

To construct the monitoring scheme, we first clustered the sensors. Figure 10 shows the hierarchical clustering dendrogram of the 26 sensors with the y axis as the cophenetic distance between different clusters. Based on the dendrogram, we divided the 26 sensors into several clusters to ensure that the minimum correlation between two sensors in the same cluster was larger than 0.2. Finally, 16 clusters were achieved as

$$\{23, (20, 21), 25, (3, 4), 24, 19, (6, 7, 9, 10), (5, 8), 1, 26, 22, 2, 18, 11, (14, 16, 17), (12, 13, 15)\},$$

where sensors inside each pair of parentheses belong to the same cluster and sensors without parentheses are individual clusters. By reordering the sensors, the strong within-cluster correlations and weak between-cluster correlations are clearly shown in Figure 11. We also considered two other extreme scenarios with $G = 1$ or $G = 26$ for mere comparison, not

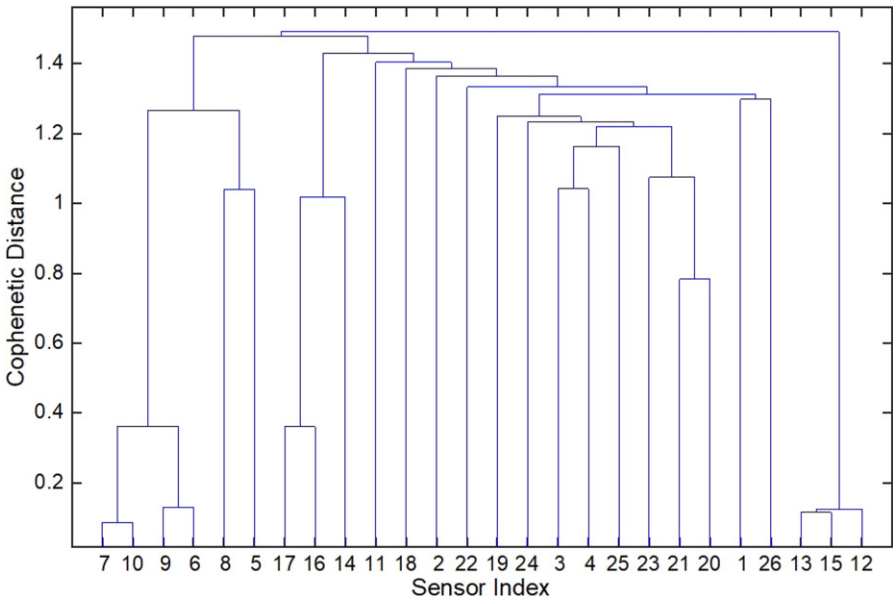

Figure 10. Hierarchical clustering for sensors.

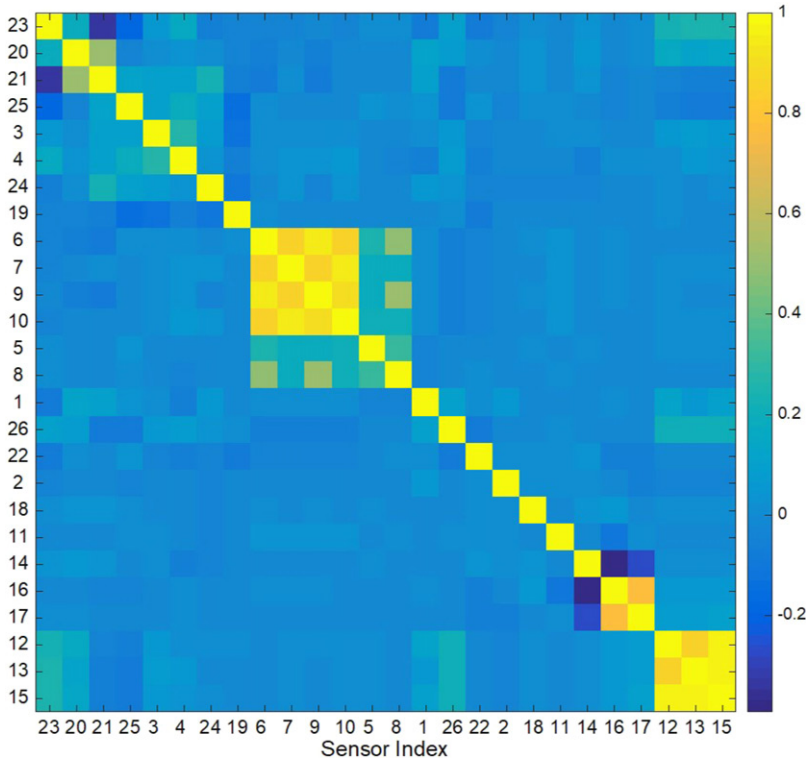

Figure 11. The correlation structure of reordered sensors based on the clustering result.

recommendation. In particular, the chart with $G = 1$ treats all sensors as one cluster, which is the exact monitoring scheme based on MFPCA (an extension of Paynabar, Qiu, and Zou 2016 to Phase II). The chart with $G = 26$ treats every single sensor as a cluster, which is the monitoring scheme based on FPCA for every sensor separately (an extension of Yu, Zou, and Wang 2012 to Phase II). As to the number of R in the monitoring scheme, according to the engineering domain knowledge we knew that the maximum number of changed sensors is not larger than 10. Therefore, we set $R = \{1, 2, 3, 6, 8, 10, 26\}$ separately for $G = 26$, $R = \{1, 2, 3, 6, 16\}$ separately for $G = 16$, and $R = 1$ for $G = 1$. In particular, $R = 1$ corresponded to the maximum chart, and $R = G$ corresponded to the sum chart. Furthermore, we also

compared our clustering-based monitoring schemes with three other state-of-the-art methods for multichannel profile monitoring, including the uncorrelated multilinear PCA (UMPCA)–based monitoring scheme (Paynabar, Jin, and Pacella 2013), the multilinear PCA (MPCA)–based monitoring scheme (Grasso, Colosimo, and Pacella 2014), and the vectorized PCA (VPCA)–based monitoring scheme (Nomikos and MacGregor 1995).

To evaluate the chart performance we purposely generated some OC patterns with different shift magnitudes from the IC samples as OC samples for testing. These generated OC patterns mimicked the true anomaly patterns in the manufacturing process. Specifically, we considered the following four types of OC shift patterns:

- **Mean shift** in sensor $j \in \{5, 6, 7, 8, 9, 10, 16, 17\}$ of magnitude $\delta \times \left(\sum_{t=1}^{T} \hat{\sigma}_{jt}/T\right)$ in the entire time duration (i.e., $\mu_{1,j}(t) = \mu_{0,j}(t) + \delta \times (\sum_{t=1}^{T} \hat{\sigma}_{jt}/T)$ for $t \in (0, 58]$) where $\hat{\sigma}_{jt}$ is the estimated standard division of the jth sensor at t.
- **Slope shift** in sensor $j \in \{14, 16, 17\}$ of magnitude $\delta \times s_j(t)$ in the entire time duration (i.e., $\mu_{1,j}(t) = \mu_{0,j}(t) + \delta \times s_j(t)$ for $t \in (0, 58]$) where $s_j(t)$ is a prespecified slope change pattern for the jth sensor at time point t. Here we determine $s_j(t)$ from real OC samples in the process.
- **Spike shift** in sensor $j \in \{2, 14, 16, 17\}$ at the time point $t = 51$ of magnitude $\delta \times (\sum_{i=1}^{46} ||\boldsymbol{Y}_{ij}||^2 /46)$ (i.e., $\mu_{1,j}(51) = \mu_{0,j}(51) + \delta \times (\sum_{i=1}^{46} ||\boldsymbol{Y}_{ij}||^2 /46)$) where $||\boldsymbol{Y}_{ij}||^2$ is the square of the l_2 norm of the jth sensor in the whole time duration for the ith IC sample.
- **Joint shift** in sensor $j \in \{5, 6, 7, 8, 9, 10, 16, 17\}$: fluctuation in sensor $j \in \{5, 7, 9, 16, 17\}$ in the time interval $t \in [20, 58]$ of magnitude δ (i.e., $\mu_{1,ij}(t) = \mu_{0,ij}(t) + s_{ij}(t)$) where $s_{ij}(t) \sim \mathrm{N}(0, \delta \times (\sum_{t=20}^{58} \hat{\sigma}_{jt}/39))$; constant mean shift in sensor $j \in \{6, 8, 10\}$ in the time interval $t \in [20, 58]$ of magnitude $\delta \times (\sum_{t=20}^{58} \hat{\sigma}_{jt}/39)$ (i.e., $\mu_{1,j}(t) = \mu_{0,j}(t) + \delta \times (\sum_{t=20}^{58} \hat{\sigma}_{jt}/39)$).

Illustrations of these generated OC patterns are shown in Figure 12. (It should be noted that the shift sizes δ in the figure are magnified more than those in the simulation, for simple illustration purpose.) We considered two EWMA parameters, $w = 0.1$ and $w = 0.05$, separately. For each w, we tuned the corresponding control limits, h_T and h_Q, for every chart to make their $ARL_0 = 200$. Then we evaluated the OC performance of different charts by simulation. In particular, in every simulation replication we randomly drew $m_0 = 100$ IC samples with replacement as reference samples from the 46 IC samples, then we drew the subsequent samples from the generated OC samples sequentially as on-line testing samples. The monitoring scheme ran until an OC alarm was generated, and the corresponding run length of this replication was recorded. This procedure was replicated 2,000 times, and the estimated *OC ARL* and the standard deviation of run length (*SDRL*) for different charts are reported in Table 1 and Table 2 for $w = 0.1$, and Table 3 and Table 4 for $w = 0.05$.

We can see that, in general, the charts with $G = 16$ perform the best for detecting these four change patterns, followed by those with $G = 26$, while the charts with $G = 1$ have the poorest results. As stated previously, this is because the charts with $G = 1$ fail to capture features of different sensor clusters since they assume all sensors come from the same cluster. On the contrary, by treating every profile separately the charts with $G = 26$, though they successfully capture different features from each sensor, are unable to capture sensor cross-correlations. Finally, we observe that the proposed charts with $G = 16$ can make a tradeoff between $G = 1$ and $G = 26$ to achieve the best results. Furthermore, among the charts with $G = 16$, those with R close to the number of clusters that include the changed sensors give the best results, such as $R = 3$ for the mean shifts and the joint shifts, $R = 1$ for the slope shifts, and $R = 2$ for the spike shifts. This is because the charts with a smaller R (such as the maximum chart in the extreme case) fail to select all the changed sensor(s). However, the charts with a larger R (such as the sum chart) may dilute the detection power by involving too many unchanged sensors. This phenomenon also occurs in the charts with $G = 1$ or $G = 26$. In practice, we may adjust R according to the domain knowledge and to achieve the optimal detection power, which demonstrates the flexibility and efficiency of the top-R rule. As to the other three charts: without considering sensor feature difference (i.e., MPCA, VPCA, and UMPCA) they perform very unsatisfactorily. In particular, for the slope shift and joint shift, all these three charts loss their detection power. For the other two shifts, though MPCA performs comparably better than the other two charts, its detection power is still much worse than our proposed method. Furthermore, UMPCA generally has the worst performance among all the charts. This is due to its inherent limitation that the maximum number of PCs that can be extracted is no more than

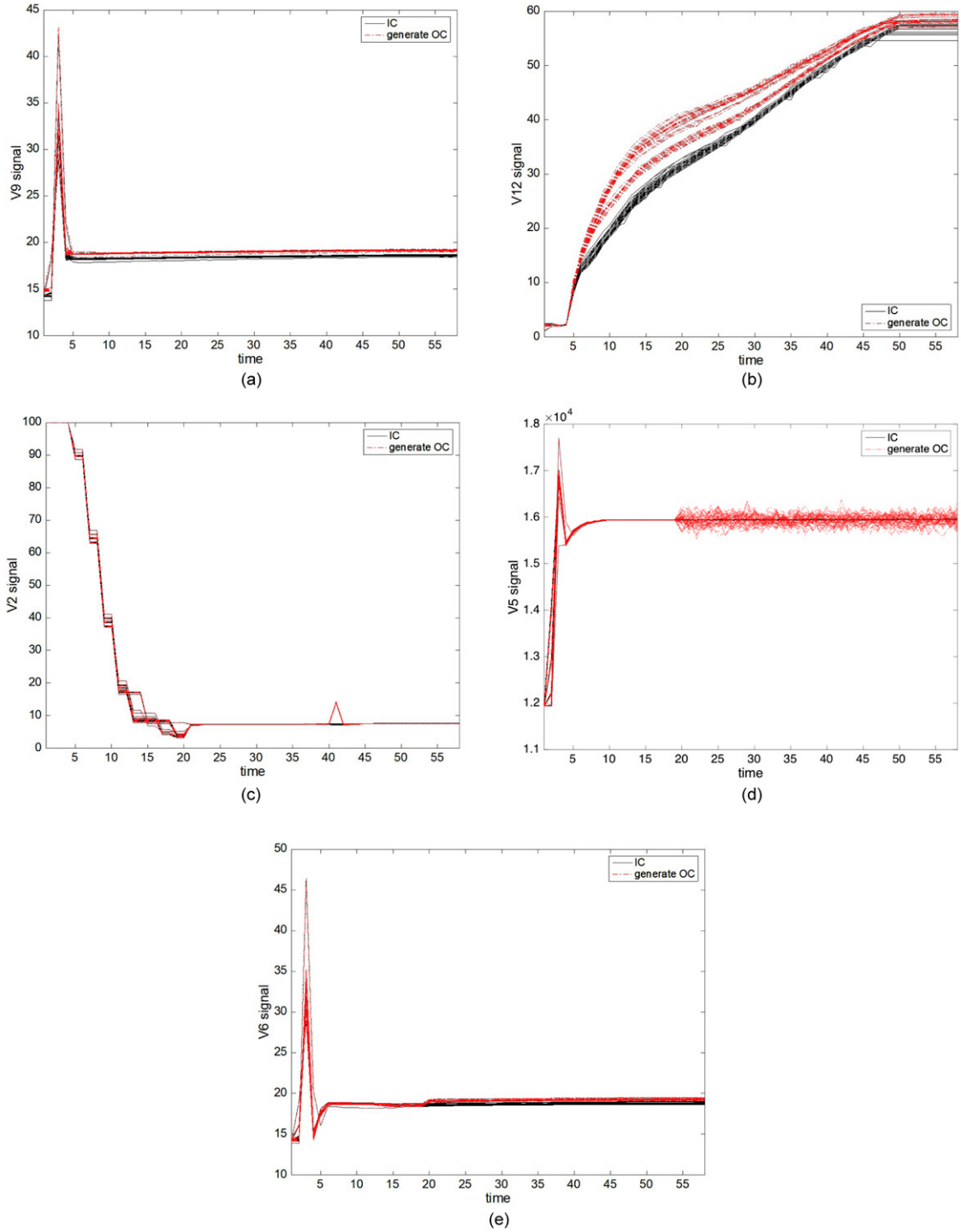

Figure 12. Illustration of the generated OC patterns. (a) Mean shift of V9 with $\delta = 5$ (b) Slope shift of V12 with $\delta = 0.2$ (c) Spike shift of V2 with $\delta = 0.2$ (d) Joint shift: fluctuation of V5 with $\delta = 5$ (e) Joint shift: mean shift of V6 with $\delta = 5$.

$\min\{g^k, m_0, T$. As a result, the explained data variation by UMPCA is very limited, rendering its poor performance for modeling and detection.

Furthermore, all the charts with $w = 0.05$ have better detection power for smaller shifts, while all the charts with $w = 0.1$ perform better for larger shifts. This is consistent with the performance of general EWMA-type charts. In practice, the selection of w mainly depends on the expected true mean shift that we would like to detect.

5. Concluding remarks

Though anomaly detection in manufacturing has been studied extensively in the literature, the following three challenges associated with efficient real-time

Table 1. *OC ARL* and *SDRL* in detecting slope shifts and mean shifts when $m_0 = 100$ and $w = 0.1$.

		G = 26 (FPCA)							G = 16					G = 1 (MFPCA)	Other methods		
	δ	R = 1	R = 2	R = 3	R = 6	R = 8	R = 10	R = 26	R = 1	R = 2	R = 3	R = 6	R = 16	R = 1	MPCA	VPCA	UMPCA
IC	0	**204**	**197**	**200**	**206**	**204**	**207**	**193**	**197**	**193**	**197**	**199**	**196**	**193**	**200**	**197**	**200**
		165	159	161	158	157	157	151	162	159	159	157	150	158	197	178	177
slope	0.01	**32.8**	**18.2**	**19.8**	**31.1**	**34.9**	**29.6**	**41.2**	**24.7**	**52.1**	**89.9**	**91.0**	**97.0**	**194**	**222**	**188**	**187**
		29.1	*10.2*	*11.5*	*23.9*	*29.1*	*20.7*	*30.9*	*58.0*	*60.9*	*103.8*	*25.1*	*28.4*	*155*	*201*	*176*	*171*
	0.02	**5.14**	**4.55**	**4.36**	**4.90**	**5.81**	**5.73**	**6.90**	**4.63**	**6.51**	**7.05**	**8.80**	**12.7**	**190**	**206**	**174**	**203**
		2.05	*1.79*	*1.79*	*1.97*	*2.43*	*1.93*	*2.22*	*2.39*	*3.28*	*3.79*	*2.45*	*3.47*	*153*	*195*	*163*	*178*
	0.03	**2.38**	**2.11**	**2.11**	**2.64**	**2.69**	**2.55**	**3.43**	**2.33**	**2.66**	**3.08**	**3.87**	**5.61**	**179**	**203**	**160**	**183**
		0.86	*0.83*	*0.81*	*1.08*	*1.05*	*0.79*	*0.89*	*1.02*	*1.29*	*1.50*	*1.27*	*1.65*	*151*	*196*	*158*	*168*
	0.04	**1.46**	**1.46**	**1.36**	**1.59**	**1.76**	**1.70**	**2.23**	**1.47**	**1.68**	**1.96**	**2.34**	**3.28**	**87.7**	**199**	**108**	**178**
		0.66	*0.61*	*0.54*	*0.65*	*0.62*	*0.60*	*0.65*	*0.68*	*0.83*	*1.01*	*0.56*	*0.78*	*89.6*	*195*	*117*	*170*
	0.05	**1.08**	**1.08**	**1.08**	**1.22**	**1.31**	**1.24**	**1.63**	**1.12**	**1.27**	**1.33**	**1.69**	**2.23**	**23.4**	**197**	**84.1**	**189**
		0.28	*0.27*	*0.29*	*0.45*	*0.53*	*0.43*	*0.56*	*0.34*	*0.52*	*0.50*	*0.52*	*0.84*	*14.3*	*188*	*99.8*	*174*
mean	0.5	**32.1**	**16.0**	**13.5**	**13.0**	**13.7**	**14.7**	**17.9**	**13.5**	**12.0**	**12.6**	**15.0**	**18.3**	**191**	**180**	**192**	**179**
		21.6	*6.74*	*5.19*	*4.81*	*5.17*	*6.03*	*7.01*	*5.12*	*4.61*	*5.40*	*7.25*	*8.30*	*153*	*181*	*178*	*167*
	1	**4.98**	**3.70**	**3.59**	**3.52**	**3.55**	**3.71**	**4.79**	**3.59**	**3.34**	**3.29**	**3.63**	**4.87**	**133**	**79.6**	**174**	**187**
		1.51	*0.89*	*0.84*	*0.83*	*0.84*	*0.74*	*1.21*	*0.87*	*0.90*	*0.93*	*1.18*	*1.44*	*129*	*101*	*169*	*172*
	1.5	**2.70**	**1.91**	**1.89**	**1.88**	**1.87**	**1.87**	**2.55**	**1.91**	**1.88**	**1.84**	**1.87**	**2.56**	**18.2**	**28.2**	**153**	**153**
		0.61	*0.29*	*0.31*	*0.33*	*0.34*	*0.33*	*0.61*	*0.29*	*0.33*	*0.37*	*0.47*	*0.65*	*13.2*	*23.8*	*156*	*155*
	2	**1.89**	**1.00**	**1**	**1**	**1**	**1.01**	**1.75**	**1**	**1**	**1**	**1.19**	**1.72**	**8.38**	**13.7**	**141**	**134**
		0.32	*0.05*	*0*	*0*	*0*	*0.08*	*0.43*	*0*	*0*	*0*	*0.39*	*0.45*	*3.46*	*7.98*	*150*	*147*
	2.5	**1**	**1**	**1**	**1**	**1**	**1**	**1.01**	**1**	**1**	**1**	**1**	**1.15**	**5.40**	**8.93**	**118**	**118**
		0	*0*	*0*	*0*	*0*	*0*	*0.08*	*0*	*0*	*0*	*0*	*0.35*	*1.99*	*4.41*	*132*	*132*
	3	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**3.96**	**6.80**	**90.4**	**97.9**
		0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1.33*	*3.06*	*113*	*123*

Table 2. *OC ARL* and *SDRL* in detecting spike shifts and joint shifts when $m_0 = 100$ and $w = 0.1$.

		G = 26 (FPCA)							G = 16					G = 1 (MFPCA)	Other methods		
	δ	R = 1	R = 2	R = 3	R = 6	R = 8	R = 10	R = 26	R = 1	R = 2	R = 3	R = 6	R = 16	R = 1	MPCA	VPCA	UMPCA
spike	0.01	**46.9**	**19.4**	**16.5**	**21.5**	**21.5**	**26.9**	**32.7**	**17.2**	**15.1**	**18.1**	**23.0**	**23.0**	**197**	**220**	**201**	**184**
		36.6	*8.16*	*6.72*	*10.4*	*10.2*	*15.3*	*19.3*	*7.0*	*6.1*	*7.8*	*13.2*	*13.2*	*159*	*201*	*182*	*169*
	0.02	**5.28**	**4.22**	**3.83**	**4.47**	**4.44**	**4.91**	**6.59**	**3.87**	**3.63**	**4.00**	**4.63**	**4.63**	**196**	**204**	**203**	**205**
		1.44	*1.13*	*1.02*	*1.24*	*1.28*	*1.38*	*1.75*	*0.94*	*0.88*	*1.27*	*1.54*	*1.54*	*6.61*	*194*	*181*	*179*
	0.03	**2.75**	**1.97**	**1.88**	**2.31**	**2.29**	**2.53**	**3.45**	**1.89**	**1.84**	**1.89**	**2.45**	**2.45**	**204**	**211**	**195**	**186**
		0.57	*0.41*	*0.33*	*0.65*	*0.65*	*0.66*	*0.73*	*0.32*	*0.36*	*0.43*	*0.72*	*0.72*	*2.10*	*198*	*176*	*170*
	0.04	**1.87**	**1.55**	**1.34**	**1.62**	**1.64**	**1.72**	**2.01**	**1.28**	**1**	**1.43**	**1.69**	**1.69**	**192**	**202**	**193**	**182**
		0.34	*0.50*	*0.48*	*0.49*	*0.48*	*0.45*	*0.42*	*0.45*	*0*	*0.50*	*0.46*	*0.46*	*0.93*	*197*	*176*	*170*
	0.05	**1**	**1**	**1**	**1**	**1**	**1**	**1.74**	**1**	**1**	**1**	**1**	**1**	**175**	**217**	**203**	**181**
		0	*0*	*0*	*0*	*0*	*0*	*0.44*	*0*	*0*	*0*	*0*	*0*	*0.59*	*194*	*178*	*172*
joint	0.5	**195**	**191**	**190**	**200**	**196**	**191**	**165**	**183**	**173**	**190**	**84.0**	**71.4**	**171**	**218**	**185**	**188**
		163	*155*	*155*	*156*	*156*	*191*	*140*	*158*	*153*	*157*	*146*	*75.8*	*146*	*197*	*177*	*171*
	1	**36.2**	**19.8**	**17.8**	**21.6**	**22.0**	**25.1**	**26.6**	**10.2**	**12.0**	**27.4**	**7.7**	**10.3**	**94.5**	**169**	**181**	**203**
		22.3	*8.70*	*7.39*	*10.6*	*11.0*	*25.1*	*15.7*	*3.8*	*5.5*	*24.9*	*24.5*	*5.6*	*105*	*174*	*166*	*181*
	1.5	**8.46**	**6.91**	**6.51**	**6.30**	**5.91**	**6.25**	**8.69**	**3.90**	**4.38**	**2.47**	**2.58**	**4.06**	**25.9**	**118**	**153**	**182**
		2.66	*2.34*	*2.12*	*2.15*	*2.24*	*6.25*	*3.30*	*1.20*	*1.57*	*2.05*	*4.08*	*1.87*	*27.1*	*140*	*152*	*167*
	2	**4.42**	**3.40**	**2.68**	**2.03**	**1.83**	**2.10**	**3.94**	**1.87**	**1.58**	**1.01**	**1.02**	**1.78**	**6.47**	**66.4**	**110**	**180**
		1.35	*1.37*	*1.32*	*0.90*	*0.74*	*2.10*	*1.49*	*0.50*	*0.58*	*0.10*	*1.56*	*0.74*	*6.16*	*78.2*	*121*	*173*
	2.5	**1.73**	**1.02**	**1**	**1**	**1**	**1**	**1.76**	**1**	**1**	**1**	**1**	**1.01**	**1.37**	**37.1**	**72.8**	**183**
		1.02	*0.16*	*0*	*0*	*0*	*1*	*0.72*	*0*	*0.04*	*0*	*0.68*	*0.09*	*0.94*	*34.2*	*84.1*	*169*
	3	**1.02**	**1**	**1**	**1**	**1**	**1**	**1.01**	**1**	**1**	**1**	**1**	**1**	**1**	**20.6**	**50.5**	**177**
		0.13	*0*	*0*	*0*	*0*	*1*	*0.09*	*0*	*0*	*0*	*0.03*	*0*	*0*	*12.7*	*54.8*	*170*

monitoring for manufacturing process variables have yet to be addressed due to large system variation, complex sensor correlation structure, different sensor features, and sparse OC change patterns. This article presented a monitoring framework to fill in this gap. Specifically, we first proposed data preprocessing methods to remove system-inherent variations, including the product fabrication time unsynchronization and manufacturing equipment long-term drift. Then, to deal with sensor feature difference, we divided the sensors into several clusters according to their correlation structures. In this way, sensors in the same cluster share common features and sensors in different clusters have different features. Then, for each cluster we used MFPCA to extract the common features and construct local monitoring statistics based on the MFPCA scores and residuals, so to detect local changes in each cluster. Collectively, with the top-*R*

Table 3. *OC* ***ARL*** and *SDRL* in detecting slope shifts and mean shifts when $m_0 = 100$ and $w = 0.05$.

		G = 26 (FPCA)							G = 16					G = 1 (MFPCA)	Other methods		
	δ	R = 1	R = 2	R = 3	R = 6	R = 8	R = 10	R = 26	R = 1	R = 2	R = 3	R = 6	R = 16	R = 1	MPCA	VPCA	UMPCA
IC	0	**204**	**197**	**201**	**202**	**201**	**202**	**201**	**198**	**203**	**199**	**196**	**198**	**202**	**201**	**201**	**202**
		158	165	164	164	175	174	182	167	168	165	163	156	158	199	186	180
slope	0.01	**12.4**	**9.42**	**11.8**	**13.9**	**17.1**	**15.4**	**20.6**	**11.6**	**18.5**	**22.5**	**27.7**	**38.7**	**190**	**204**	**195**	**198**
		6.55	*4.85*	*5.94*	*7.40*	*9.27*	*8.50*	10.9	*8.71*	*13.5*	*16.7*	*24.9*	*32.0*	*159*	*194*	*180*	*174*
	0.02	**3.78**	**3.35**	**3.33**	**3.68**	**3.90**	**4.83**	**6.39**	**2.98**	**4.13**	**4.15**	**5.10**	**6.90**	**96.6**	**202**	**165**	**195**
		1.67	*1.39*	*1.44*	*1.63*	*1.66*	*2.00*	*2.49*	*1.86*	*3.17*	*3.10*	*3.89*	*4.42*	*111*	*193*	*159*	*181*
	0.03	**1.93**	**1.66**	**1.69**	**2.03**	**2.10**	**2.14**	**3.09**	**1.41**	**1.94**	**2.04**	**2.04**	**2.72**	**11.6**	**208**	**117**	**191**
		0.73	*0.67*	*0.73*	*0.88*	*0.94*	*0.94*	*1.39*	*0.65*	*1.22*	*137*	*1.33*	*1.79*	*14.2*	*194*	*117*	*170*
	0.04	**1.13**	**1.12**	**1.18**	**1.39**	**1.66**	**1.47**	**1.80**	**1.08**	**1.26**	**1.30**	**1.61**	**1.98**	**6.45**	**197**	**69.4**	**175**
		0.37	*0.34*	*0.41*	*0.57*	*0.69*	*0.94*	*0.72*	*0.27*	*0.55*	*0.57*	*0.78*	*1.13*	*8.67*	*190*	*72.8*	*173*
	0.05	**1.04**	**1.02**	**1.10**	**1.15**	**1.16**	**1.28**	**1.33**	**1.03**	**1.03**	**1.11**	**1.13**	**1.37**	**3.36**	**189**	**45.5**	**174**
		0.06	*0.14*	*0.32*	*0.38*	*0.36*	*0.48*	*0.50*	*0.17*	*0.17*	*0.33*	*0.39*	*0.56*	*4.91*	*185*	*46.6*	*171*
mean	0.5	**13.0**	**10.0**	**9.51**	**9.60**	**9.74**	**9.83**	**12.2**	**12.3**	**10.4**	**10.0**	**11.4**	**13.9**	**176**	**121**	**94.4**	**185**
		6.30	*4.55*	*3.84*	*3.74*	*3.94*	*3.98*	*4.66*	*6.11*	*4.62*	*4.75*	*5.55*	*6.45*	*160*	*137*	*102.3*	*170*
	1	**3.87**	**3.14**	**2.78**	**2.74**	**2.83**	**2.90**	**3.62**	**3.54**	**3.28**	**3.03**	**3.26**	**3.91**	**150**	**39.7**	**27.3**	**180**
		1.33	*0.99*	*0.66*	*0.67*	*0.81*	*0.85*	*1.09*	*1.26*	*1.03*	*0.99*	*1.26*	*1.49*	*154*	*34.8*	*22.7*	*168*
	1.5	**1.87**	**1.81**	**1.80**	**1.73**	**1.73**	**1.67**	**1.84**	**1.81**	**1.78**	**1.72**	**1.76**	**2.09**	**82.5**	**18.5**	**13.0**	**130**
		0.34	*0.39*	*0.39*	*0.44*	*0.45*	*0.46*	*0.47*	*0.39*	*0.42*	*0.45*	*0.53*	*0.69*	*125*	*10.9*	*8.60*	*136*
	2	**1.40**	**1**	**1**	**1**	**1**	**1**	**1.33**	**1.19**	**1**	**1**	**1.16**	**1.53**	**23.4**	**11.5**	**8.17**	**107**
		0.49	*0*	*0*	*0*	*0*	*0*	*0.47*	*0.40*	*0*	*0*	*0.36*	*0.50*	*48.1*	*3.83*	*4.97*	*127*
	2.5	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1.04**	**11.5**	**7.87**	**5.38**	**89.5**
		0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0.19*	*8.68*	*3.83*	*3.09*	*110*
	3	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**8.03**	**5.67**	**4.31**	**66.3**
		0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*5.77*	*2.55*	*2.37*	*91.6*

Table 4. *OC ARL* and *SDRL* in detecting spike shifts and joint shifts when $m_0 = 100$ and $w = 0.05$.

		G = 26 (FPCA)							G = 16					G = 1 (MFPCA)	Other methods		
	δ	R = 1	R = 2	R = 3	R = 6	R = 8	R = 10	R = 26	R = 1	R = 2	R = 3	R = 6	R = 16	R = 1	MPCA	VPCA	UMPCA
spike	0.01	**14.4**	**10.8**	**10.6**	**10.9**	**11.8**	**12.4**	**16.2**	**11.5**	**10.1**	**10.8**	**13.2**	**16.3**	**12.9**	**199**	**206**	**192**
		7.03	*5.08*	*4.40*	*4.66*	*5.22*	*5.60*	*6.49*	*4.68*	*4.51*	*5.06*	*6.17*	*6.92*	*6.47*	*189*	*187*	*173*
	0.02	**4.21**	**3.30**	**3.14**	**3.26**	**3.40**	**3.60**	**4.70**	**3.18**	**3.06**	**3.16**	**3.64**	**4.47**	**3.96**	**192**	**201**	**198**
		1.33	*1.06*	*0.99*	*1.01*	*1.09*	*1.19*	*1.37*	*1.05*	*0.99*	*1.09*	*1.40*	*1.51*	*1.35*	*185*	*185*	*182*
	0.03	**2.11**	**1.77**	**1.73**	**1.74**	**1.74**	**1.78**	**2.48**	**1.79**	**1.77**	**1.72**	**1.84**	**2.43**	**2.12**	**191**	**198**	**190**
		0.60	*0.42*	*0.44*	*0.451*	*0.44*	*0.47*	*0.68*	*0.41*	*0.42*	*0.44*	*0.57*	*0.72*	*0.68*	*186*	*178*	*173*
	0.04	**1.65**	**1.01**	**1**	**1.01**	**1.18**	**1.33**	**1.72**	**1**	**1**	**1**	**1.38**	**1.67**	**1.53**	**202**	**197**	**187**
		0.48	*0.08*	*0*	*0.07*	*0.38*	*0.47*	*0.45*	*0*	*0*	*0*	*0.48*	*0.47*	*0.49*	*185*	*180*	*178*
	0.05	**1**	**1**	**1**	**1**	**1**	**1**	**1.05**	**1**	**1**	**1**	**1**	**1.10**	**1**	**196**	**189**	**179**
		0	*0*	*0*	*0*	*0*	*0*	*0.22*	*0*	*0*	*0*	*0*	*0.29*	*0*	*186*	*175*	*168*
joint	0.5	**153**	**111**	**120**	**116**	**121**	**116**	**105**	**158**	**134**	**34.6**	**100**	**108**	**148**	**178**	**196**	**195**
		145	*112*	*113*	*118*	*118*	*114*	*101*	*153*	*133*	*35.8*	*107*	*96.2*	*144*	*177*	*179*	*173*
	1	**13.4**	**10.9**	**10.9**	**9.6**	**11.8**	**12.1**	**14.7**	**10.6**	**11.2**	**5.59**	**12.8**	**15.0**	**31.7**	**111**	**164**	**199**
		6.06	*4.94*	*4.66*	*5.00*	*5.40*	*5.54*	*7.18*	*5.12*	*5.56*	*3.34*	*7.33*	*8.51*	*38.8*	*123*	*155*	*182*
	1.5	**6.29**	**5.00**	**4.81**	**4.16**	**4.11**	**4.02**	**5.75**	**4.30**	**4.41**	**1.91**	**4.59**	**6.03**	**8.44**	**57.9**	**110**	**192**
		2.45	*2.07*	*1.91*	*1.69*	*1.74*	*1.76*	*2.55*	*1.75*	*1.97*	*0.86*	*2.44*	*3.10*	*7.49*	*52.6*	*117*	*171*
	2	**3.26**	**2.11**	**1.54**	**1.36**	**1.31**	**1.34**	**2.30**	**2.27**	**1.94**	**1.65**	**1.97**	**2.88**	**2.94**	**30.4**	**67.4**	**177**
		1.44	*1.13*	*0.82*	*0.47*	*0.48*	*0.51*	*1.04*	*0.72*	*0.65*	*0*	*0.89*	*1.40*	*2.61*	*18.8*	*72.3*	173
	2.5	**1.14**	**1.01**	**1**	**1**	**1**	**1**	**1.12**	**1.08**	**1**	**1**	**1.02**	**1.48**	**1.11**	**20.5**	**45.4**	**180**
		0.41	*0.12*	*0*	*0*	*0*	*0*	*0.32*	*0.27*	*0*	*0*	*0.14*	*0.59*	*0.41*	*10.5*	*49.5*	*170*
	3	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**14.4**	**30.6**	**164**
		0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*6.83*	*25.1*	*166*

rule, these local monitoring statistics are fused together as a global monitoring statistic that can have efficient and scalable detection power for sparse OC patterns. Finally, a case study with real data from a manufacturing process demonstrated the efficacy of the proposed framework in terms of OC ARL.

Along with this direction, there are still several valuable extensions. First, the current hierarchical clustering algorithm treats every time point of a profile as an independent observation without considering within-profile correlations. It is desirable to construct a more advanced clustering algorithm taking within-profile correlations into account. Second, the current monitoring framework considers two separate steps—sensor clustering and data fusion—to deal with different sensor features. However, in some cases it might not be so straightforward or distinguishable to identify different clusters. With this in mind, it is of

interest to construct a uniform monitoring scheme that can handle different sensor features directly. Last but not least, diagnostics is equally important as change detection. As such, we may consider developing a diagnostic procedure for the current monitoring scheme, after an OC alarm is triggered.

About the authors

Dr. Zhang is an Assistant Professor in the Department of Industrial Engineering at Tsinghua University. Her email is zhangchen01@tsinghua.edu.cn.

Dr. Yan is an Assistant Professor in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. His email is HaoYan@asu.edu.

Dr. Lee is a principal engineer at Samsung Electronics. His email is mountlee@gmail.com.

Dr. Shi is the Carolyn J. Stewart Chair and Professor in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology. He is the corresponding author. His email is jianjun.shi@isye.gatech.edu.

Acknowledgments

The authors are grateful for the valuable comments provided by the editors and referees.

Funding

This article is partiallysupported by the NSF Grant ID 1233143.

References

Ding, Y., L. Zeng, and S. Zhou. 2006. Phase I analysis for monitoring nonlinear profiles in manufacturing processes. *Journal of Quality Technology* 38 (3):199–216. doi:10.1080/00224065.2006.11918610

Ghashghaei, R., and A. Amiri. 2016. Sum of squares control charts for monitoring of multivariate multiple linear regression profiles in phase II. *Quality and Reliability Engineering International* 33 (4):767–784. doi:10.1002/qre.2055.

Grasso, M., B. M. Colosimo, and M. Pacella. 2014. Profile monitoring via sensor fusion: The use of PCA methods for multi-Channel Data. *International Journal of Production Research* 52 (20):6110–35. doi:10.1080/00207543.2014.916431

Jackson, J. E., and G. S. Mudholkar. 1979. Control procedures for residuals associated with principal component analysis. *Technometrics* 21 (3):341–9. doi:10.1080/00401706.1979.10489779

Jensen, W. A., J. B. Birch, and W. H. Woodall. 2008. Monitoring correlation within linear profiles using mixed models. *Journal of Quality Technology* 40 (2):167–83. doi:10.1080/00224065.2008.11917723

Jin, J., and J. Shi. 1999. Feature-preserving data compression of stamping tonnage information using wavelets. *Technometrics* 41 (4):327–39. doi:10.1080/00401706.1999.10485932

Jin, J., and J. Shi. 2001. Automatic feature extraction of waveform signals for in-process diagnostic performance improvement. *Journal of Intelligent Manufacturing* 12 (3):257–68. doi:10.1023/A:1011248925750

Keogh, E., and C. A. Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7 (3):358–86. doi:10.1007/s10115-004-0154-9

Kim, J., Q. Huang, J. Shi, and T. S. Chang. 2006. Online multichannel forging tonnage monitoring and fault pattern discrimination using principal curve. *Journal of Manufacturing Science and Engineering* 128 (4):944–50.

Lee, S.-P., A.-K. Chao, F. Tsung, D. S. H. Wong, S.-T. Tseng, and S.-S. Jang. 2011. Monitoring batch processes with multiple on-off steps in semiconductor manufacturing. *Journal of Quality Technology* 43 (2):142–57. doi:10.1080/00224065.2011.11917852

Liu, K., Y. Mei, and J. Shi. 2015. An adaptive sampling strategy for online high-dimensional process monitoring. *Technometrics* 57 (3):305–19. doi:10.1080/00401706.2014.947005

Mahmoud, M. A., and W. H. Woodall. 2004. Phase I analysis of linear profiles with calibration applications. *Technometrics* 46 (4):380–91. doi:10.1198/004017004000000455

Mei, Y. 2010. Efficient scalable schemes for monitoring a large number of data streams. *Biometrika* 97 (2):419–33. doi:10.1093/biomet/asq010

Mei, Y. 2011. Quickest detection in censoring sensor networks. Paper presented at the 2011 IEEE International Symposium on Information Theory, St. Petersburg, Russia, July 31- August 5.

Nomikos, P., and J. F. MacGregor. 1995. Multivariate SPC charts for monitoring batch processes. *Technometrics* 37 (1):41–59. doi:10.1080/00401706.1995.10485888

Noorossana, R., M. Eyvazian, and A. Vaghefi. 2010. Phase II monitoring of multivariate simple linear profiles. *Computers & Industrial Engineering* 58 (4):563–70. doi:10.1016/j.cie.2009.12.003

Noorossana, R., M. Eyvazian, A. Amiri, and M. A. Mahmoud. 2010. Statistical monitoring of multivariate multiple linear regression profiles in phase I with calibration application. *Quality and Reliability Engineering International* 26 (3):291–303. doi:10.1002/qre.1066

Noorossana, R., A. Saghaei, and A. Amiri. 2011. *Statistical analysis of profile monitoring*. Vol. 865. Hoboken, NJ: John Wiley & Sons.

Paynabar, K., and J. Jin. 2011. Characterization of non-linear profiles variations using mixed-effect models and wavelets. *IIE Transactions* 43 (4):275–90. doi:10.1080/0740817X.2010.521807

Paynabar, K., J. Jin, and M. Pacella. 2013. Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis. *IIE Transactions* 45 (11):1235–47. doi:10.1080/0740817X.2013.770187

Paynabar, K., P. Qiu, and C. Zou. 2016. A change point approach for phase-I analysis in multivariate profile

monitoring and diagnosis. *Technometrics* 58 (2):191–204. doi:10.1080/00401706.2015.1042168

Qiu, P., C. Zou, and Z. Wang. 2010. Nonparametric profile monitoring by mixed effects modeling. *Technometrics* 52 (3):265–77. doi:10.1198/TECH.2010.08188

Tartakovsky, A. G., B. L. Rozovskii, R. B. Blažek, and H. Kim. 2006. Detection of intrusions in information systems by sequential change-point methods. *Statistical Methodology* 3 (3):252–93. doi:10.1016/j.stamet.2005.05.003

Thorndike, R. L. 1953. Who belongs in the family? *Psychometrika* 18 (4):267–76. doi:10.1007/BF02289263

Tibshirani, R., G. Walther, and T. Hastie. 2001. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (2):411–23. doi:10.1111/1467-9868.00293

Walker, E., and S. P. Wright. 2002. Comparing curves using additive models. *Journal of Quality Technology* 3(1):118–29. doi:10.1080/00224065.2002.11980134

Woodall, W. H. 2007. Current research on profile monitoring. *Production* 17 (3):420–5. doi:10.1590/S0103-65132007000300002

Yu, G., C. Zou, and Z. Wang. 2012. Outlier detection in functional observations with applications to profile monitoring. *Technometrics* 54 (3):308–18. doi:10.1080/00401706.2012.694781

Zhang, C., N. Chen, and C. Zou. 2016. Robust multivariate control chart based on goodness-of-fit test. *Journal of Quality Technology* 48 (2):139–61. doi:10.1080/00224065.2016.11918156

Zhou, S., B. Sun, and J. Shi. 2006. An SPC monitoring system for cycle-based waveform signals using HAAR transform. *IEEE Transactions on Automation Science and Engineering* 3 (1):60–72.

Zou, C., F. Tsung, and Z. Wang. 2007. Monitoring general linear profiles using multivariate exponentially weighted moving average schemes. *Technometrics* 49 (4):395–408. doi:10.1198/004017007000000164

Zou, C., F. Tsung, and Z. Wang. 2008. Monitoring profiles based on nonparametric regression methods. *Technometrics* 50 (4):512–26. doi:10.1198/004017008000000433

Zou, C., X. Ning, and F. Tsung. 2012. LASSO-based multivariate linear profile monitoring. *Annals of Operations Research* 192 (1):3–19. doi:10.1007/s10479-010-0797-8