Taylor & Francis
Taylor & Francis Group

Check for updates

Transparent Sequential Learning for Statistical Process Control of Serially Correlated Data

Peihua Qiu and Xiulin Xie

Department of Biostatistics, University of Florida, Gainesville, FL

ABSTRACT

Machine learning methods have been widely used in different applications, including process control and monitoring. For handling statistical process control (SPC) problems, conventional supervised machine learning methods (e.g., artificial neural networks and support vector machines) would have some difficulties. For instance, a training dataset containing both in-control and out-of-control (OC) process observations is required by a supervised machine learning method, but it is rarely available in SPC applications. Furthermore, many machine learning methods work like black boxes. It is often difficult to interpret their learning mechanisms and the resulting decision rules in the context of an application. In the SPC literature, there have been some existing discussions on how to handle the lack of OC observations in the training data, using the one-class classification, artificial contrast, real-time contrast, and some other novel ideas. However, these approaches have their own limitations to handle SPC problems. In this article, we extend the self-starting process monitoring idea that has been employed widely in modern SPC research to a general learning framework for monitoring processes with serially correlated data. Under the new framework, process characteristics to learn are well specified in advance, and process learning is sequential in the sense that the learned process characteristics keep being updated during process monitoring. The learned process characteristics are then incorporated into a control chart for detecting process distributional shift based on all available data by the current observation time. Numerical studies show that process monitoring based on the new learning framework is more reliable and effective than some representative existing machine learning SPC approaches.

1. Introduction

Statistical process control (SPC) provides a powerful tool for online process monitoring in the manufacturing industry, internet traffic management, disease surveillance, and many other applications (Hawkins and Olwell 1998; Montgomery 2012; Qiu 2014). One major task of SPC is to detect distributional shifts in certain quality characteristics from an in-control (IC) distribution, and give signals once such shifts are detected. The SPC problem is often challenging due to complexity of the observed process data. This article aims to tackle the challenging SPC problem by developing a general sequential learning framework.

In the SPC literature, there have been many existing methods for various process monitoring applications. Early SPC research is designed mainly for monitoring production lines in the manufacturing industry under the conventional assumptions that process observations at different observation times are independent and identically distributed with a common parametric distribution (e.g., normal distribution) when the related process is IC (e.g., Crosier 1998; Hawkins, Qiu, and Kang 2003; Lowry et al. 1992). In the past two decades, SPC has found many new applications in public health, environmental science, business management, and other industries and disciplines. The conventional model assumptions mentioned above are rarely valid in these applications. For instance, the observed daily cases of the influenza-like illness at different clinics and hospitals in the United States would be both spatially and temporally correlated, and their distribution can hardly be described well by a parametric distribution (e.g., Yang and Qiu 2020). Thus, in recent SPC research, some more flexible control charts have been developed for handling cases when the IC process distribution does not have a parametric form (e.g., Chakraborti and Graham 2019; Qiu 2018), or when process observations are serially correlated (e.g., Capizzi and Masarotto 2008; Prajapati and Singh 2012; Xue and Qiu 2020). Recent overviews of different SPC charts, especially those related to big data and Industry 4.0, can be found in Qiu (2020) and Reis and Gins (2017).

In the recent years, machine learning methods have been under rapid development (e.g., Aggarwal 2018; Hastie, Tibshirani, and Friedman 2001). A conventional machine learning method tries to approximate an application by a computer algorithm after learning the data structure of the related problem from a training dataset. An attractive feature of such methods is their flexibility, in the sense that they usually do not impose restrictive model assumptions explicitly. For this reason, machine learning methods have been used widely in different applications, including SPC. An SPC problem

is essentially a sequential classification problem, in which the status of a process should be classified into either the IC or the out-of-control (OC) status at each time during process monitoring. To solve a classification problem by a supervised machine learning method, a training dataset containing observations of both classes (e.g., IC and OC) is required, and then a classification rule can be developed by a computer algorithm (e.g., random forest (RF)) from the training dataset. However, in many SPC applications, we only have an IC dataset available beforehand, which is often collected after a proper Phase I process monitoring (cf., Jones-Farmer et al. 2014), and no OC process observations would be available in advance. Thus, it is difficult to apply a conventional supervised machine learning method to the SPC problem directly.

To overcome the difficulty mentioned above, Tuv and Runger (2003) suggested the so-called *artificial contrast* approach, consisting of the following two steps: (i) an artificial dataset was generated from a specific off-target distribution (e.g., a uniform distribution) and observations in this dataset were regarded as OC data, and (ii) a classification rule was then developed from the training data, which combine the IC data and the artificial OC data, by a regular supervised machine learning algorithm (e.g., RF). See Hu, Runger, and Tuv (2007), Hwang, Runger, and Tuv (2007), and Li, Runger, and Tuv (2006) for some modifications and generalizations. Of course, the artificial OC data generated in the above method may not represent the actual off-target process observations well in a given application. To overcome this limitation, Deng, Runger, and Tuv (2012) suggested the so-called *real-time contrast (RTC)* method, in which previous process observations in a window of the current observation time were treated as OC data, they were combined with the IC data to form a training dataset, and then a classification rule was developed accordingly from the training dataset. An alternative strategy to employ machine learning algorithms for SPC is to use the so-called *one-class classification (OCC)* procedures, first discussed in Sun and Tsung (2003). OCC is based on the support vector data description (SVDD) approach in the computer science literature (Tax and Duin 2004), and its main idea is to envelop a one-class training dataset with the volume as small as possible. Then, the boundary found by OCC can be used as a classification rule. In the context of SPC, the IC dataset obtained before online process monitoring can be used as the one-class training dataset. Then, a future process observation can be judged as IC if it is within the boundary of the training dataset found by OCC, and OC otherwise. There are several different OCC-based control charts. See, for instance, He, Jiang, and Deng (2018), Sukchotrat, Kim, and Tsung (2010), and the references cited therein. For recent overviews on control charts constructed based on machine learning approaches, see Megahed and Jones-Farmer (2015), Weese et al. (2016), and Zhang, Tsung, and Zou (2015).

A machine learning method is often considered as a "black box" in describing the observed data of a process in the sense that it is usually difficult to interpret its learning mechanism and the statistical properties of the resulting decision rule. Its performance also depends heavily on how well a training dataset rep-

resents the population to study. These features are shared by the control charts discussed above that are based on machine learning algorithms. For instance, in an artificial contrast approach, the artificial OC process observations generated from a specific parametric (e.g., uniform) distribution may not represent actual OC process observations well, which could cause unsatisfactory performance of the related control chart. Therefore, such SPC charts based on the machine learning approaches have much room for improvement.

In an SPC application, an IC dataset is routinely collected after a Phase I analysis for estimating certain IC parameters, and then the estimated IC parameters could be used in constructing a Phase II control chart for online process monitoring (Qiu 2014, p. 7). By comparing this common practice of process monitoring in the SPC literature with the OCC-based control charts discussed above, we can see that the two strategies share some similarities in that an IC dataset is used for estimating the IC process distribution and the estimated IC process distribution is then used in the decision-making at each time point during online process monitoring. In practice, the IC dataset is often quite small. Consequently, a relatively large variability in the estimated IC parameters could negatively affect the performance of a Phase-II control chart. To overcome this limitation, Hawkins (1987) suggested a self-starting chart, by which process observations at a given time point are combined with the IC data if the process is declared to be IC at that time point, and then the IC parameter estimates can be updated using the combined IC dataset for process monitoring at the next time point. Compared to the conventional SPC charts, the self-starting chart can alleviate the dependence of its performance on the IC data due to the updating mechanism for the IC parameter estimates, although it has been demonstrated that its IC performance still has certain variability due to the randomness of the IC data (Keefe, Woodall, and Jones-Farmer 2015). This self-starting process monitoring idea has been widely used in the SPC literature (e.g., Hawkins and Maboudou-Tchao 2007, Zou, Zhou, and Wang 2007). Based on this idea, we develop a general learning framework in this article for online process monitoring. In the new framework, the data structure to learn is well specified, or it is "transparent" with regard to what should be learned about the data structure. The new learning process is sequential in the sense that its training data keep being expanded and its learned IC data structure keeps being updated during online process monitoring. Because of these features, the new learning method is referred to as *transparent sequential learning (TSL)*. Under this framework, the learned IC data structure is then used in constructing a Phase II control chart for online process monitoring. The proposed TSL-based process monitoring scheme can be used for online monitoring of both univariate and multivariate numerical processes with serially correlated observations. The related quality variables could be mutually correlated, and have a nonparametric distribution. The idea of TSL-based process monitoring is actually general. After proper modifications, it should be able to solve other SPC problems, including dynamic process monitoring and profile monitoring, which is left for future research.

Compared to the existing machine learning control charts discussed above, the proposed TSL-based process monitoring

scheme has at least the following strengths. First, it is flexible enough to handle processes with serially correlated data, while the existing machine learning control charts are designed mainly for monitoring processes with independent data. Second, because the TSL-based process monitoring scheme is based on the self-starting idea and its IC data keep being expanded before a signal is given, it can provide a reliable online process monitoring with a relatively small initial IC data. As a comparison, the existing machine learning control charts usually require a quite large training data to have a reliable performance. Third, the data structure to learn from the IC data is well-defined in the TSL-based process monitoring scheme, and the related chart would give a signal once it detects a difference between the structure of the newly observed data and the learned structure of the IC data. This feature of transparent learning could make the related chart effective since the learned data structure has been used in the construction of the chart and a shift in any component of such structure would trigger a signal. As a comparison, the existing machine learning control charts are more generic and would be less sensitive to a shift in a component of the defined data structure. These intuitive observations will be confirmed later by numerical studies. In the SPC literature, many self-starting charts have been developed for handling various process monitoring tasks (Qiu 2014, secs. 4.5 and 5.4). However, as far as we know, there is limited existing discussion on self-starting monitoring of certain processes, including multivariate processes with nonstationary serial data correlation. Such a challenging SPC problem can be handled properly by the proposed TSL-based process monitoring scheme.

The remaining parts of the article are organized as follows. In Section 2, the proposed learning framework TSL and the TSL-based process monitoring scheme are described in detail. Numerical performance of the TSL-based process monitoring scheme is investigated in Section 3 by several simulation studies. A real-data example to demonstrate the application of the proposed method is discussed in Section 4. Some remarks conclude the article in Section 5. To save space, some materials are presented in a supplementary file.

2. Process Monitoring by Transparent Sequential Learning

In this section, the proposed learning framework TSL and the TSL-based process monitoring scheme are described in detail. It should be pointed out that the current TSL-based process monitoring scheme is described mainly for Phase II online monitoring of processes whose quality characteristics are observed at the same equally spaced times, after the related processes being adjusted properly by Phase I SPC methods. In practice, Phase I SPC is an indispensable step to adjust and control a process at its initial stage. See related discussions in Capizzi and Masarotto (2013), Jones-Farmer et al. (2014), and the references cited therein.

2.1. Specific Process Characteristics to Learn and the TSL Framework

Assume that $\mathbf{X} = (X_1, X_2, \ldots, X_p)'$ is a vector of $p \geq 1$ numerical quality characteristics to monitor about a sequential process. Its observation at time n is denoted as $\mathbf{X}_n = (X_{n1}, X_{n2}, \ldots, X_{np})'$. To online monitor the sequential process $\{\mathbf{X}_n, n \geq 1\}$, an IC dataset $\mathcal{X}_{IC} = \{\mathbf{X}_{-m_0+1}, \mathbf{X}_{-m_0+2}, \ldots, \mathbf{X}_0\}$ of size m_0 is assumed to be available in advance. Our major goal is to detect a distributional shift as soon as possible during online process monitoring. Once a signal of shift is triggered by a control chart, process experts need to make a judgment whether the detected shift is practically important and figure out its root causes if the answer is positive (cf., Sall 2018).

Process characteristics to learn: Data structure of the process $\{\mathbf{X}_n, n \geq 1\}$ can be described by (i) the distribution of $\mathbf{X}_n$ at each n, and (ii) the serial data correlation among observations $\{\mathbf{X}_n, n \geq 1\}$ at different time points. When the process is IC, each observation is assumed to follow the IC process distribution, denoted as $F_0(\mathbf{x})$, for any $\mathbf{x} \in R^p$, and the IC serial data correlation can be described by $\text{cov}(\mathbf{X}_i, \mathbf{X}_{i'})$, for any $i, i' \geq 1$. Thus, the main process characteristics for TSL to learn from the observed data include $F_0(\mathbf{x})$ and $\text{cov}(\mathbf{X}_i, \mathbf{X}_{i'})$.

Obviously, there are so many unknown quantities in $F_0(\mathbf{x})$ and $\text{cov}(\mathbf{X}_i, \mathbf{X}_{i'})$, and it may not be realistic to estimate all of them from the observed data. One reasonable solution is to develop strategies to simplify these quantities in a way that the learning task becomes manageable but the restriction on the IC data structure is still flexible enough to cover many process monitoring applications. In this article, we present the TSF learning framework in a special case briefly described below. In many applications it is reasonable to assume that the correlation between two process observations gradually decreases when their observation times get farther away. This assumption can be approximated by the assumption that $\boldsymbol{\gamma}_n(s) := \text{cov}(\mathbf{X}_n, \mathbf{X}_{n-s}) = \mathbf{0}$ when $s \geq b_{\max}$ and $n \geq 1$, where $b_{\max}$ is a parameter denoting the time range of serial data correlation. Here, $\boldsymbol{\gamma}_n(s)$ is allowed to depend on n (i.e., the serial data correlation could be nonstationary). In some applications, if the serial data correlation can be assumed to be stationary, then $\boldsymbol{\gamma}_n(s)$ would not depend on n. In such cases, $\boldsymbol{\gamma}_n(s)$ can be simply written as $\boldsymbol{\gamma}(s)$. In this article, we will discuss both cases when the serial data correlation is stationary or nonstationary. For the IC distribution $F_0(\mathbf{x})$, we are often concerned about its mean $\boldsymbol{\mu}_0$ and covariance matrix Σ_0 in the SPC literature since these quantities are routinely used for measuring the quality/performance of the related process. In the special case described above, the major IC process characteristics for TSL to learn from the observed data are restricted to $\boldsymbol{\mu}_0$, Σ_0 and $\{\boldsymbol{\gamma}_n(s), 0 \leq s \leq b_{\max}, n \geq 1\}$.

The TSL framework: Our proposed learning framework TSL works as follows. The IC parameters are initially estimated from the IC data $\mathcal{X}_{IC}$, and the initial estimates are then used for online process monitoring. At the current time point during sequential online process monitoring, if the TSL-based control chart claims that the related process is IC, then the IC parameter estimates get updated properly after the observed data at the current time point are combined with the IC data. In that way, TSL keeps learning from the sequentially collected process observations about the IC process characteristics until a shift is detected by a TSL-based control chart.

Initial parameter estimates: To estimate the IC parameters from the IC data $\mathcal{X}_{IC}$, there are two general methods. One is based on the maximum likelihood estimation, and the other

is based on the moment estimation. Because the first method requires parametric forms to describe the IC process distribution and the IC serial data correlation which are unavailable here, the moment estimation is considered here. In cases when the serial data correlation is stationary, the moment estimates of the IC parameters are

$$\widehat{\boldsymbol{\mu}}_0^{(0)} = \frac{1}{m_0} \sum_{i=-m_0+1}^{0} \mathbf{X}_i$$

$$\widehat{\boldsymbol{\gamma}}^{(0)}(s) = \frac{1}{m_0 - s} \sum_{i=-m_0+1}^{-s} \left(\mathbf{X}_{i+s} - \widehat{\boldsymbol{\mu}}_0^{(0)} \right) \left(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_0^{(0)} \right)',$$
$$\text{for } 0 \le s \le b_{\max}. \quad (1)$$

Because $\Sigma_0 = \boldsymbol{\gamma}(0)$, the initial estimate of Σ_0 is $\widehat{\Sigma}_0^{(0)} = \widehat{\boldsymbol{\gamma}}^{(0)}(0)$. In cases when the serial data correlation is nonstationary, the covariance matrices $\{\boldsymbol{\gamma}_n(s), 0 \le s \le b_{\max}, -m_0 + 1 \le n \le 0\}$ can be estimated by the following weighted moment estimates: for $-m_0 + 1 \le n \le 0$ and $0 \le s \le b_{\max}$,

$$\widehat{\boldsymbol{\gamma}}_n(s) = \frac{\sum_{i=-m_0+1}^{-s} \left(\mathbf{X}_{i+s} - \widehat{\boldsymbol{\mu}}_0^{(0)} \right) \left(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_0^{(0)} \right)' K\left(\min(|i+s-n|, |i-n|)/g \right)}{\sum_{i=-m_0+1}^{-s} K\left(\min(|i+s-n|, |i-n|)/g \right)}, \quad (2)$$

where $K(u) = \frac{3}{4}(1 - u^2)I(|u| \le 1)$ is the Epanechnikov kernel function, and g is a bandwidth. In Equation (2), $\widehat{\boldsymbol{\gamma}}_n(s)$ is a weighted average of all terms $(\mathbf{X}_{i+s} - \widehat{\boldsymbol{\mu}}_0^{(0)})(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_0^{(0)})'$, in which at least one of $i+s$ and i is in the neighborhood $(n-g, n+g)$ and the weight is determined by the kernel function. In the kernel smoothing literature, the Epanechnikov kernel function is often used because of its good theoretical properties (Epanechnikov 1969). The bandwidth g can be chosen by minimizing the following cross-validated prediction error (PE):

$$\text{PE}(g) = \frac{1}{m_0} \sum_{i=-m_0+1}^{0} \left(\mathbf{X}_i - \widehat{\mathbf{X}}_{-i} \right)' \left(\mathbf{X}_i - \widehat{\mathbf{X}}_{-i} \right),$$

where $\widehat{\mathbf{X}}_{-i}$ is the predicted value of $\mathbf{X}_i$ obtained by the kriging method (Cressie 1990) described below. For $-m_0 + 1 \le i \le 0$, let $\mathbf{Y}_{-i} = (\mathbf{X}_{\max(-m_0+1, i-g)}, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_{\min(0, i+g)})$ be the matrix of IC observations with their indices in the neighborhood $[i-g, i+g]$ and with $\mathbf{X}_i$ excluded, and $\widehat{\mathbf{e}}_{-i} = \mathbf{Y}_{-i} - (\widehat{\boldsymbol{\mu}}_0^{(0)}, \ldots, \widehat{\boldsymbol{\mu}}_0^{(0)}, \widehat{\boldsymbol{\mu}}_0^{(0)}, \ldots, \widehat{\boldsymbol{\mu}}_0^{(0)})$ be the corresponding matrix of residuals. Then, the predicted value $\widehat{\mathbf{X}}_{-i}$ is defined to be $\widehat{\mathbf{X}}_{-i} = \widehat{\boldsymbol{\mu}}_0^{(0)} + \widehat{V}'_{i,-i} \widehat{V}_{-i}^{-1} \widehat{\mathbf{e}}_{-i}$, where $\widehat{V}_{i,-i}$ is the estimated covariance matrix between $\mathbf{X}_i$ and $\mathbf{Y}_{-i}$, $\widehat{V}_{-i}$ is the estimated covariance matrix of $\mathbf{Y}_{-i}$, and both of them can be computed from $\{\widehat{\boldsymbol{\gamma}}_n(s)\}$ defined in Equation (2).

2.2. TSL-Based Online Process Monitoring

In this part, we discuss TSL-based online monitoring of the process observations $\{\mathbf{X}_n, n \ge 1\}$ in cases when they are serially correlated with the correlation coefficients $\{\boldsymbol{\gamma}_n(s), 0 \le s \le b_{\max}, n \ge 1\}$. In the SPC literature, most conventional control charts are designed for cases when process observations are assumed to be serially uncorrelated. So, at the current time point n, after the process observation $\mathbf{X}_n$ is obtained, it needs to be decorrelated with all previous process observations before a control chart is applied. Under the TSL learning framework, if the chart declares the process to be IC at n, then $\mathbf{X}_n$ can be combined with the IC dataset and the estimates of the IC parameters can get updated from the combined IC dataset. Otherwise, the chart gives a signal of shift. The entire TSL-based online process monitoring procedure is demonstrated in Figure 1, and its major components are described briefly below.

Data decorrelation: At the current time point n, we would like to decorrelate the observation $\mathbf{X}_n$ with all previous observations $\{\mathbf{X}_i, i \le n - 1\}$. Because of the assumption that two process observations are uncorrelated if their observation times are more than $b_{\max}$ apart, we only need to decorrelate $\mathbf{X}_n$ with $\{\mathbf{X}_i, n - b_{\max} \le i \le n - 1\}$. Since data decorrelation needs to be executed at each observation time during online process monitoring, reduction of computing time is important. To this end, You and Qiu (2019) suggested using the *spring length* concept that was originally proposed in Chatterjee and Qiu (2009). Assume that a CUSUM chart is used in the proposed TSL-based process monitoring scheme. Then, the spring length T_n at time n is defined to be the number of observation times between n and the last time when the CUSUM charting statistic is reset to zero. Because the CUSUM chart has the restarting mechanism that all process observations collected before the time $n-T_n$ are all ignored in the subsequent process monitoring, $\mathbf{X}_n$ only needs to be decorrelated with observations collected at the previous $b_n = \min\{T_{n-1}, b_{\max}\}$ time points, where T_{n-1}, instead of T_n, is used here since T_n is unavailable yet before a decision is made about the process status at time n. Due to the fact that T_{n-1} is often a single-digit integer when the process is IC, the computing time can be reduced substantially by using it here. Furthermore, it has been shown in You and Qiu (2019) that (i) a distributional shift in the original data would be attenuated by data decorrelation and consequently a control chart applied to the decorrelated data could be less sensitive to the shift, and (ii) the use of spring length can alleviate this "masking effect" of data decorrelation since the spring length is often a small number and thus only a small number of previous observations are involved in data decorrelation at each time. A recursive algorithm for data decorrelation modified from the one in You and Qiu (2019) is briefly described below.

Let $\mathbf{W}_n = (\mathbf{X}'_{n-b_n}, \mathbf{X}'_{n-b_n+1}, \ldots, \mathbf{X}'_n)'$ be a long vector consisting of $\mathbf{X}_n$ and all previous observations that it needs to be decorrelated with. Then, its estimated IC covariance matrix is

$$\widehat{\Sigma}_{b_n+1, b_n+1}(n) = \begin{pmatrix} \widehat{\boldsymbol{\gamma}}_{n-b_n-1}(0) & \cdots & \widehat{\boldsymbol{\gamma}}_{n-1}(b_n) \\ \vdots & \ddots & \vdots \\ [\widehat{\boldsymbol{\gamma}}_{n-1}(b_n)]' & \cdots & \widehat{\boldsymbol{\gamma}}_{n-1}(0) \end{pmatrix}$$

$$=: \begin{pmatrix} \widehat{\Sigma}_{b_n, b_n}(n) & \widehat{\boldsymbol{\sigma}}_{b_n, n} \\ (\widehat{\boldsymbol{\sigma}}_{b_n, n})' & \widehat{\boldsymbol{\gamma}}_{n-1}(0) \end{pmatrix},$$

where $\widehat{\boldsymbol{\sigma}}_{b_n, n} = ([\widehat{\boldsymbol{\gamma}}_{n-1}(b_n)]', \ldots, [\widehat{\boldsymbol{\gamma}}_{n-1}(1)]')'$ and $\{\widehat{\boldsymbol{\gamma}}_i(s), 0 \le s \le b_n, n - b_n - 1 \le i \le n - 1\}$ are estimates of $\{\boldsymbol{\gamma}_i(s), 0 \le s \le$
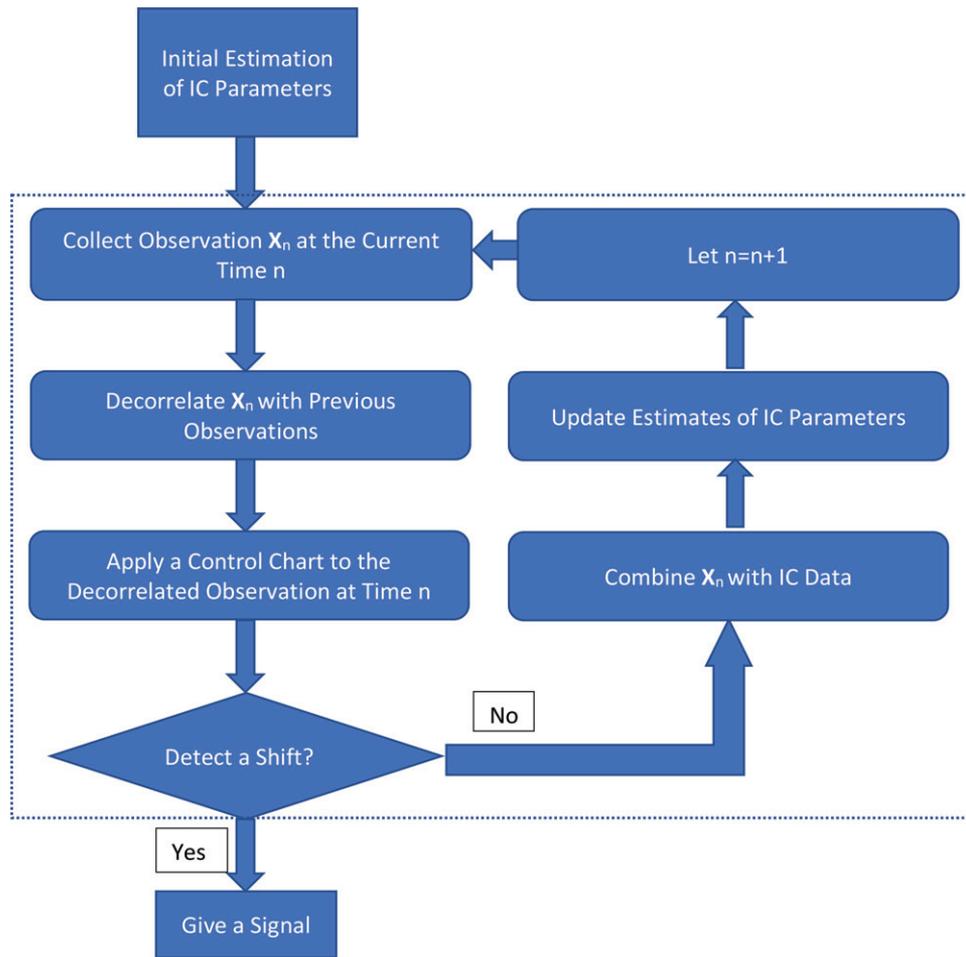
Figure 1. Diagram of the proposed TSL-based online process monitoring scheme. The dotted rectangle highlights the TSL learning framework.

$b_n, n - b_n - 1 \leq i \leq n - 1\}$ defined in Equation (6). Here, the IC covariance of $\mathbf{W}_n$ is estimated by the estimated covariance coefficients obtained at time $n - 1$ because it is unknown yet whether the underlying process is IC at time n during the data decorrelation at time n. If $b_n = 0$, then we can define the standardized observation at time n to be $[\widehat{\boldsymbol{\gamma}}_{n-1}(0)]^{-1/2}(\mathbf{X}_n - \widehat{\boldsymbol{\mu}}_0^{(n-1)})$ and it does not need to be decorrelated with any previous observations, where $\widehat{\boldsymbol{\mu}}_0^{(n-1)}$ is defined in Equation (5). Otherwise, based on the Cholesky decomposition of $\widehat{\Sigma}_{b_n+1,b_n+1}(n)$, it can be checked that $\widehat{D}_{b_n}^{-1/2}(n)[\mathbf{X}_n - \widehat{\boldsymbol{\mu}}_0^{(n-1)} - (\widehat{\boldsymbol{\sigma}}_{b_n,n})'\widehat{\Sigma}_{b_n,b_n}^{-1}(n)\widehat{\mathbf{e}}_{n-1}]$ would have the asymptotic identity covariance matrix and be asymptotically uncorrelated with $\mathbf{X}_{n-b_n}, \mathbf{X}_{n-b_n+1}, \ldots, \mathbf{X}_{n-1}$,

where $\widehat{D}_{b_n}(n) = \widehat{\boldsymbol{\gamma}}_{n-1}(0) - (\widehat{\boldsymbol{\sigma}}_{b_n,n})'\widehat{\Sigma}_{b_n,b_n}^{-1}(n)\widehat{\boldsymbol{\sigma}}_{b_n,n}$ and $\widehat{\mathbf{e}}_{n-1} = [(\mathbf{X}_{n-b_n} - \widehat{\boldsymbol{\mu}}_0^{(n-1)})', (\mathbf{X}_{n-b_n+1} - \widehat{\boldsymbol{\mu}}_0^{(n-1)})', \ldots, (\mathbf{X}_{n-1} - \widehat{\boldsymbol{\mu}}_0^{(n-1)})']'$. Then, the decorrelated and standardized observation at time n can be defined to be

$$\mathbf{X}_n^* = \begin{cases} [\widehat{\boldsymbol{\gamma}}_{n-1}(0)]^{-1/2}\left(\mathbf{X}_n - \widehat{\boldsymbol{\mu}}_0^{(n-1)}\right), & \text{when } b_n = 0, \\ \\ \widehat{D}_{b_n}^{-1/2}(n)\left[\mathbf{X}_n - \widehat{\boldsymbol{\mu}}_0^{(n-1)} - (\widehat{\boldsymbol{\sigma}}_{b_n,n})'\widehat{\Sigma}_{b_n,b_n}^{-1}(n)\widehat{\mathbf{e}}_{n-1}\right], \\ & \text{when } b_n > 0. \end{cases}$$

(3)

In Equation (3), the inverse matrix $\widehat{\Sigma}_{b_n,b_n}^{-1}(n)$ can be computed recursively by the following formula: for $2 \leq i \leq b_n$,

$$\widehat{\Sigma}_{i,i}^{-1}(n) = \begin{pmatrix} \widehat{\Sigma}_{i-1,i-1}^{-1}(n) + \widehat{\Sigma}_{i-1,i-1}^{-1}(n)\widehat{\boldsymbol{\sigma}}_{i-1,n}\widehat{D}_{i-1}^{-1}(n)(\widehat{\boldsymbol{\sigma}}_{i-1,n})'\widehat{\Sigma}_{i-1,i-1}^{-1}(n), & -\widehat{\Sigma}_{i-1,i-1}^{-1}(n)\widehat{\boldsymbol{\sigma}}_{i-1,n}\widehat{D}_{i-1}^{-1}(n) \\ -\widehat{D}_{i-1}^{-1}(n)(\widehat{\boldsymbol{\sigma}}_{i-1,n})'\widehat{\Sigma}_{i-1,i-1}^{-1}(n), & \widehat{D}_{i-1}^{-1}(n) \end{pmatrix}.$$

After using the above data decorrelation algorithm, the resulting decorrelated process observations are denoted as $\{\mathbf{X}_n^* = (X_{n1}^*, X_{n2}^*, \ldots, X_{np}^*)', n \geq 1\}$. In cases when the serial data correlation is stationary, $\widehat{\Sigma}_{b_n+1,b_n+1}(n)$ can be computed from $\{\widehat{\boldsymbol{\gamma}}^{(n-1)}(s), 0 \leq s \leq b_n\}$ defined in Equation (5).

Apply a control chart to the decorrelated data: First, note that the transformed observation $\mathbf{X}_n^*$ in Equation (3), for each n, is a linear combination of $\mathbf{X}_n$ and its previous b_n observations. So, the distribution of $\mathbf{X}_n^*$ would be closer to a normal distribution when $b_n > 1$, compared to the distribution of $\mathbf{X}_n$, due to

the central limit theorem. So, if we are certain the distribution of the original observations $\{\mathbf{X}_n\}$ is close to normal, then we can consider using a conventional multivariate control chart for monitoring the decorrelated data. For instance, to use the multivariate EWMA (MEWMA) chart by Lowry et al. (1992), let us define

$$\mathbf{E}_n = \lambda(\mathbf{X}_n^* - \widehat{\boldsymbol{\mu}}_0^{(0)}) + (1 - \lambda)\mathbf{E}_{n-1}, \text{ for } n \geq 1, \qquad (4)$$

where $\mathbf{E}_0 = \mathbf{0}$, $\lambda \in (0, 1]$ is a weighting parameter, and $\widehat{\boldsymbol{\mu}}_0^{(0)}$ is the estimated IC mean obtained from the initial IC data. Then, the chart gives a signal when

$$\mathbf{E}_n' \widehat{\Sigma}_{\mathbf{E}_n}^{-1} \mathbf{E}_n > h,$$

where $\widehat{\Sigma}_{\mathbf{E}_n} = [\lambda/(2 - \lambda)]\widehat{\Sigma}_0^{(0)}$, $\widehat{\Sigma}_0^{(0)}$ is the estimated IC covariance matrix obtained from the initial IC data, and $h > 0$ is a control limit. The self-starting version of the MEWMA chart (4), denoted as MEWMA-SS, can be obtained by replacing $\widehat{\boldsymbol{\mu}}_0^{(0)}$ and $\widehat{\Sigma}_0^{(0)}$ in the above formulas by $\widehat{\boldsymbol{\mu}}_0^{(n-1)}$ and $\widehat{\Sigma}_0^{(n-1)}$ defined in Equation (5), respectively.

In practice, the value of b_n is usually small, as mentioned earlier. Thus, the distribution of $\mathbf{X}_n^*$ could be substantially different from normal in cases when the distribution of the original observation $\mathbf{X}_n$ is substantially different from normal. In such cases, we suggest using a nonparametric chart. In the SPC literature, there are many nonparametric charts developed (Qiu 2018), and most of them can be used here. In this article, we demonstrate the proposed TSL-based process monitoring approach using the nonparametric chart based on data categorization that was proposed by Qiu (2008). Construction of this chart is described in the supplementary file. It is denoted as TSL-S when the serial data correlation is assumed stationary, and TSL-NS when the serial data correlation is allowed to be non-stationary.

Update the IC parameter estimates: When a TSL-based control chart does not give a signal at time n, under the TSL learning framework, the observation $\mathbf{X}_n$ should be combined with the IC dataset and the estimates of the IC parameters $\boldsymbol{\mu}_0$, Σ_0 and $\{\boldsymbol{\gamma}_n(s), 0 \leq s \leq b_{\max}, n \geq 1\}$ should be updated using the combined IC dataset. To reduce computation, it is critically important to update the IC parameter estimates recursively when it is possible. To this end, the following recursive formulas can be derived when the serial data correlation is assumed stationary: for $n \geq 1$ and $0 \leq s \leq b_{\max}$,

$$\widehat{\boldsymbol{\mu}}_0^{(n)} = \frac{1}{m_0 + n}\mathbf{X}_n + \frac{m_0 + n - 1}{m_0 + n}\widehat{\boldsymbol{\mu}}_0^{(n-1)}, \qquad (5)$$

$$\widehat{\boldsymbol{\gamma}}^{(n)}(s) = \frac{1}{m_0 + n - s}\left(\mathbf{X}_n - \widehat{\boldsymbol{\mu}}_0^{(n)}\right)\left(\mathbf{X}_{n-s} - \widehat{\boldsymbol{\mu}}_0^{(n)}\right)'$$
$$+ \frac{m_0 + n - s - 1}{m_0 + n - s}\widehat{\boldsymbol{\gamma}}^{(n-1)}(s).$$

In Equation (5), the quantities $\widehat{\boldsymbol{\mu}}_0^{(0)}$ and $\{\widehat{\boldsymbol{\gamma}}^{(0)}(s), 0 \leq s \leq b_{\max}\}$ are defined in Equation (1). Also, it is obvious that $\widehat{\Sigma}_0^{(n)} = \widehat{\boldsymbol{\gamma}}^{(n)}(0)$. After the IC parameter estimates are updated, we are ready to monitor the process at the next time point $n + 1$, as

demonstrated in Figure 1. If the serial data correlation is non-stationary, then $\{\boldsymbol{\gamma}_n(s), 0 \leq s \leq b_{\max}, n \geq 1\}$ can be estimated by

$$\widehat{\boldsymbol{\gamma}}_n(s) = \frac{\sum_{i=n-w}^{n}\left(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_0^{(n)}\right)\left(\mathbf{X}_{i-s} - \widehat{\boldsymbol{\mu}}_0^{(n)}\right)' K\left((n-i)/g\right)}{\sum_{i=n-w}^{n} K\left((n-i)/g\right)},$$
$$\text{for } 0 \leq s \leq b_{\max}, \ n \geq 1, \qquad (6)$$

where w is a prespecified window size, and $K(\cdot)$ and g are the same as those in (2). Since $\widehat{\boldsymbol{\gamma}}_n(s)$ is computed from observations with indices in a local neighborhood of n, instead of from all previous observations, its computation should be manageable. Because of this and the fact that its recursive computation can only be accomplished asymptotically, which would lose some numerical accuracy, a recursive formula to compute $\widehat{\boldsymbol{\gamma}}_n(s)$ is not provided here and we suggest using Equation (6) instead.

2.3. Practical Guidelines on Parameter Selection

On selection of m_0: Performance of a TSL-based control chart depends on the initial IC data size m_0. As shown in Table 1 in Section 3 and Table S.3 in the supplementary file, when the serial data correlation is stationary and the dimensionality p is small (e.g., $p \leq 5$), the IC performance of the charts TSL-S and TSL-NS is reliable when $m_0 \geq 400$ since their ARL_0 values would be within 10% of the nominal ARL_0 value in such cases. If the serial data correlation is non-stationary or p is relatively large, then m_0 should be chosen larger to have a reliable IC performance.

On selection of $b_{\max}$: When constructing a TSL-based control chart, the serial data correlation is assumed to vanish when two observation times are at least $b_{\max}$ apart. In practice, $b_{\max}$ is unknown. Based on our extensive numerical experience, the performance of a TSL-based control chart could not be improved much if $b_{\max}$ is chosen larger than 20, and its performance would be negatively affected in certain cases if $b_{\max}$ is chosen smaller than 10. So, we suggest choosing $b_{\max} \in [10, 20]$. In all numerical examples in Sections 3 and 4, $b_{\max}$ is chosen to be 20.

On selection of w: When the serial data correlation is non-stationary, the parameters $\{\boldsymbol{\gamma}_n(s), 0 \leq s \leq b_{\max}, n \geq 1\}$ need to be estimated from previous observations whose observation times are within a window of size w of the current observation time n (cf., (6)). Based on our numerical experience, w can be chosen to be $w = a \times b_{\max}$ with $a \in [4, 6]$. In the related numerical examples in Sections 3 and 4, a is chosen to be 5.

2.4. Some Discussions on Different Machine Learning Control Charts

From the above discussions, it can be seen that the IC process characteristics to learn are well specified in the proposed TSL-based process monitoring scheme, which include the IC process distribution $F_0(\mathbf{x})$ and the IC covariance matrices $\mathrm{cov}(\mathbf{X}_i, \mathbf{X}_{i'})$, for any $\mathbf{x} \in R^p$ and $i, i' \geq 1$. These quantities jointly describe the IC data structure of the process under monitoring. Because there are so many unknowns among them, it is still an open

research problem how to sequentially estimate them properly from the IC data. In this article, we present the TSL-based process monitoring scheme in a simplified case when $F_0(\mathbf{x})$ is replaced by its mean $\boldsymbol{\mu}_0$ and covariance matrix Σ_0 and when $\text{cov}(\mathbf{X}_i, \mathbf{X}_{i'})$ is reduced to the parameters $\{\boldsymbol{\gamma}_n(s), 0 \leq s \leq b_{\max}, n \geq 1\}$. Although this simplified case can provide a reasonable approximation to the IC data structure in some applications, there are certainly some other applications that it may not be able to handle properly. For instance, if the IC distribution $F_0(\mathbf{x})$ is skewed, then its mean and covariance matrix cannot describe the whole distribution well. There are also some applications in which the serial data correlation could be long-range (e.g., Altmann, Cristadoro, and Esposti 2012). Therefore, much future research is needed to generalize the TSL-based process monitoring scheme discussed in this article to handle more applications.

As discussed in Section 1, there are two types of existing control charts using machine learning algorithms. The first type employs the supervised learning idea. Because there are usually no OC data available before online process monitoring, some methods in the first type use artificial data that are generated from a prespecified off-target distribution (e.g., uniform), while some others treat previous process observations within a local window of the current observation time as the OC data. In both cases, the artificial data and/or the observations within a local window of the current observation time could be substantially different from the actual OC data. Consequently, the effectiveness of the decision rules generated from such subjectively generated/assigned training data could be compromised. Furthermore, the decision rules in these methods are usually generated by certain supervised machine learning algorithms (e.g., random forest), and their interpretation and statistical properties are often difficult to discuss. The second type of existing machine learning control charts employs the OCC idea. Their decision rules are obtained from the IC dataset by estimating the spatial boundary of the IC process distribution $F_0(\mathbf{x})$. Because there is often no prior information on the shape or other related features of $F_0(\mathbf{x})$, these methods often use local smoothing approaches to compute the boundary estimates (cf., Sun and Tsung 2003). Thus, the variability of the estimates would be relatively large, especially in SPC applications that the IC data sizes are usually quite small. Consequently, the related control charts would not be reliable enough to use, which is confirmed in Section 3 by numerical examples. Because these charts depend on the boundary of $F_0(\mathbf{x})$ only, they cannot detect any distributional shifts that minimally change the boundary of $F_0(\mathbf{x})$. Such shifts, however, could be important to detect since they may change the mean and/or variance of each quality characteristic under monitoring.

3. Simulation Studies

In this section, we investigate the numerical performance of the TSL-based process monitoring charts MEWMA-SS, TSL-S, and TSL-NS discussed in Section 2.2, in comparison with four representative existing process monitoring methods that are based on machine learning algorithms. The four existing methods considered here are briefly described below.

- The kernel distance based multivariate control chart suggested by Sun and Tsung (2003), denoted as KC (representing kernel-distance-based classification): The chart KC is based on an SVDD algorithm. It works roughly as follows. First, the kernel distance (kd) from the center of a training dataset can be computed for each process observation. The chart then gives a signal when kd at the current time point exceeds a control limit h, where h is determined from the training dataset such that the Type I error probability is below a given level α. The ARL_0 value of the chart is defined to be $1/\alpha$, since this method assumes that process observations at different time points are independent. In KC, Sun and Tsung (2003) suggested using the Gaussian radical basis function (RBF) as the kernel function, and the bandwidth used in RBF is chosen such that the average of the Type I and Type II error probabilities computed from the training data is minimized.

- The control chart suggested by Sukchotrat, Kim, and Tsung (2010) that is based on the K-nearest-neighbor (KNN) data description procedure, denoted as KNN: To construct the chart KNN, the average distance between a given observation and $\rho = 30$ nearest observations in the training dataset is calculated, which is called K^2-value of the given observation. Then, the $(1 - \alpha)$th percentile of the K^2-values of all observations in the training dataset is used as the control limit h. To reduce variability, Sukchotrat, Kim, and Tsung (2010) suggested determining h by the following bootstrap procedure: (i) a total of $B = 1000$ bootstrap samples are obtained from the training dataset by the random sampling procedure with replacement and each bootstrap sample has the same size as the training dataset, (ii) the $(1 - \alpha)$th percentile of the K^2-values can be computed from each bootstrap sample, as described above, and (iii) h is chosen to be the mean of the B such percentiles. Then, for online process monitoring, the process is declared to be OC at a given time if the K^2-value of the related process observation exceeds the control limit h. This method also assumes that process observations at different time points are independent. Thus, its ARL_0 value equals $1/\alpha$.

- The multivariate control chart using the artificial contrasts approach and the RF classification that was suggested by Hwang, Runger, and Tuv (2007), denoted as RF: For a given IC dataset, the same amount of artificial contrasts are generated and combined with the IC dataset to form a training dataset. For individual quality characteristics X_j, their artificial contrasts are generated independently from uniform distributions with the ranges same as those of X_j values in the IC dataset, for $j = 1, 2, \ldots, p$. A binary decision rule is then generated by the RF algorithm from the training data so that the Type I error probability is α. In the RF classification, the number of trees used is set to be 1000 and the number of random variables selected for each tree is chosen to be $\sqrt{p}$, as did in Hwang, Runger, and Tuv (2007). During online process monitoring, the process status is determined by the decision rule. Since this method assumes that process observations at different time points are independent, its ARL_0 value also equals $1/\alpha$.

- The process monitoring method using the RTC approach that was suggested by Deng, Runger, and Tuv (2012), denoted as RTC: The RTC method treats the process monitoring

problem as a real-time classification problem, in which process observations in the IC dataset and those within a moving window of the current time point form a training dataset, with the former as IC observations and the latter as OC observations. Then, a decision rule is generated from this training dataset using the RF algorithm, and the process status at the current time point is determined by this decision rule. As discussed in Deng, Runger, and Tuv (2012), there could be several possible charting statistics based on the RF algorithm. As in their simulation studies, the estimated out-of-bag correct classification rate in the IC data is used as the charting statistic in this article. By following their suggestions, the window size of all moving windows is fixed at 10, and the threshold value used in the decision rules is determined from the IC dataset by a bootstrap procedure with the bootstrap sample size of 1000 so that a given ARL_0 value is reached.

In all simulation examples, the nominal ARL_0 values of all charts are fixed at 200. So, the significance level α in the charts KC, KNN and RF is chosen to be 0.005. If there is no further specification, then the procedure parameter k in the TSL-based CUSUM charts TSL-S and TSL-NS (see the expression immediately below Expression (A.1) in the supplementary file) is chosen to be 0.01, the weighting parameter λ in the MEWMA-SS chart is fixed at 0.05, the number of nearest observations ρ in KNN is chosen to be 30 as suggested by Sukchotrat, Kim, and Tsung (2010), and the moving window size in RTC is chosen to be 10 as suggested by Deng, Runger, and Tuv (2012).

Regarding the IC process distribution and the IC serial data correlation, the following five cases when $p = 5$ are considered first:

Case I: Process observations $\{\mathbf{X}_n, n \geq 1\}$ are iid. with the IC distribution $N_5(\mathbf{0}, I_{5 \times 5})$.

Case II: Process observations $\mathbf{X}_n = (X_{n1}, X_{n2}, X_{n3}, X_{n4}, X_{n5})'$ are generated as follows: X_{n1} follows the AR(1) model $X_{n1} = 0.1 X_{n-1,1} + \epsilon_{n1}$, where $X_{01} = 0$ and $\{\epsilon_{n1}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution; X_{n2} is generated by the model $X_{n2} = X_{n1} + \epsilon_{n2}$, where $\{\epsilon_{n2}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution; X_{n3} follows the AR(2) model $X_{n3} = 0.2 X_{n-1,3} + 0.1 X_{n-2,3} + \epsilon_{n3}$, where $X_{03} = X_{13} = 0$ and $\{\epsilon_{n3}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution; X_{n4} is generated by the model $X_{n4} = X_{n3} + \epsilon_{n4}$, where $\{\epsilon_{n4}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution; X_{n5} is generated by the model $X_{n5} = 0.4 X_{n1} + 0.6 X_{n3} + \epsilon_{n5}$, where $\{\epsilon_{n5}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution. In all models discussed above, $\{\epsilon_{n1}\}$, $\{\epsilon_{n2}\}$, $\{\epsilon_{n3}\}$, $\{\epsilon_{n4}\}$ and $\{\epsilon_{n5}\}$ are independent.

Case III: Process observations $\mathbf{X}_n = (X_{n1}, X_{n2}, X_{n3}, X_{n4}, X_{n5})'$ are generated in the same way as that in Case II, except that $\{\epsilon_{n1}\}$, $\{\epsilon_{n2}\}$, $\{\epsilon_{n3}\}$, $\{\epsilon_{n4}\}$ and $\{\epsilon_{n5}\}$ are iid random errors whose common distribution is the same as that of $0.1(\xi - 3)/\sqrt{6}$, where ξ is a random variable having the distribution of χ_3^2.

Case IV: Process observations $\mathbf{X}_n = (X_{n1}, X_{n2}, X_{n3}, X_{n4}, X_{n5})'$ are generated as follows: for each j, $X_{nj} = 0.5 \xi_{nj} + \epsilon_{nj}$, where $\{\epsilon_{nj}\}$ are iid. random errors with the $N(0, 1)$

distribution, and $\{\xi_{nj}\}$ is a two-state Markov process with the initial state being 0 and the transition matrix between the two states $\{0, 1\}$ being

$$\begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}.$$

Case V: Process observations $\mathbf{X}_n = (X_{n1}, X_{n2}, X_{n3}, X_{n4}, X_{n5})'$ are generated as follows: X_{n1} follows the AR(1) model $X_{n1} = 0.01\sqrt{n} X_{n-1,1} + \epsilon_{n1}$, where $X_{01} = 0$ and $\{\epsilon_{n1}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution; X_{n2} is generated by the model $X_{n2} = X_{n1} + \epsilon_{n2}$, where $\{\epsilon_{n2}\}$ are iid random errors with the distribution of $0.1(\xi - 3)/\sqrt{6}$ and ξ is a random variable having the distribution of χ_3^2; X_{n3} follows the AR(1) model $X_{n3} = 0.1 \log(n) X_{n-1,3} + \epsilon_{n3}$, where $X_{03} = 0$ and $\{\epsilon_{n3}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution; X_{n4} is generated by the model $X_{n4} = X_{n3} + \epsilon_{n4}$, where $\{\epsilon_{n4}\}$ are iid random errors with the distribution of $0.1(\xi - 3)/\sqrt{6}$ and ξ is a random variable having the distribution of χ_3^2; and X_{n5} is generated by the model $X_{n5} = 0.1\sqrt{n} \epsilon_{n5}$, where $\{\epsilon_{n5}\}$ are iid random errors with the $N(0, 0.1^2)$ distribution. In all models discussed above, $\{\epsilon_{n1}\}$, $\{\epsilon_{n2}\}$, $\{\epsilon_{n3}\}$, $\{\epsilon_{n4}\}$ and $\{\epsilon_{n5}\}$ are independent.

In Cases I–V described above, each quality characteristic is standardized to have mean 0 and variance 1. Obviously, Case I is the conventional case considered in the SPC literature with iid process observations and the standard normal IC process distribution. Cases II and III consider correlated process observations across different quality characteristics and different observation times with stationary serial data correlation. The correlation among different quality characteristics is present only for pairs (X_{n1}, X_{n2}), (X_{n3}, X_{n4}), (X_{n1}, X_{n5}), and (X_{n3}, X_{n5}) in these two cases. The error distribution is normal in Case II and skewed in Case III. In Case IV, each component of $\{\boldsymbol{\xi}_n = (\xi_{n1}, \xi_{n2}, \xi_{n3}, \xi_{n4}, \xi_{n5})', n \geq 1\}$ is a binary Markov process. Its observations are serially correlated because the transition probability from a given state to the same state is 0.75, which is larger than the transition probability from a given state to the other state. Such a serial data correlation is stationary, but cannot be described by a conventional time series model. The process observations $\{\mathbf{X}_n\}$ are generated by adding some random noise to $\{\boldsymbol{\xi}_n\}$, and thus they are also serially correlated. In Case V, the serial data correlation depends on observation times, and thus is non-stationary.

Evaluation of the IC performance: We first evaluate the IC performance of the related control charts. In the simulation study, the IC sample size m_0 can change among $\{200, 300, 400, 500, 1000, 2000\}$. For the MEWMA-SS chart, its control limit is determined by Monte Carlo simulations based on the assumed IC normal distribution. For the TSL-S and TSL-NS charts, their control limits are determined by the bisection searching algorithm described in Section S.1 of the supplementary file. The control limits of the four competing methods KC, KNN, RF, and RTC are determined as discussed in their brief descriptions given above. For each method, after an IC dataset of size m_0 is generated, some IC parameters are estimated from the IC dataset. Then, a run length (RL) value is recorded from a

simulation of online process monitoring of 2000 IC process observations. The simulation is then repeated for 1000 times, and the conditional ARL_0 value conditional on the IC data is estimated as the average of the 1000 RL values. This entire simulation study, from generation of the IC dataset to computation of the conditional ARL_0 value, is then repeated for 100 times. The average of the 100 conditional ARL_0 values obtained from these 100 replicated simulations is then used as the final estimate of the true ARL_0 value of the related control chart, and the standard error of the estimated ARL_0 value can also be calculated. To make the comparison fair, process observations are decorrelated before each method is used for process monitoring. The decorrelation procedure discussed in Section 2.2 under the stationary assumption is used for all control charts, except TSL-NS for which non-stationary serial data correlation is allowed in the decorrelation procedure (cf., Equations (2) and (6)). The estimated ARL_0 values in various different cases considered are shown in Figure 2 here, and presented in Table S.1 in the supplementary file as well.

From Figure 2, it can be seen the following results. First, the IC performance of the charts TSL-S and TSL-NS is quite reliable in Cases I-IV when $m_0 \geq 400$ since their estimated ARL_0 values are within 10% of the nominal ARL_0 value of 200 in all cases considered, and TSL-S performs a little better in most of these cases. For Case V when the serial data correlation is non-stationary, TSL-NS is still reliable when $m_0 \geq 400$, but TSL-S is not that reliable in such cases since it is constructed based on the stationarity assumption. Second, in Cases I–IV, the charts KNN, RF, and RTC have a reasonably reliable performance when $m_0 \geq 1000$, and the chart MEWMA-SS is quite reliable when $m_0 \geq 500$. Third, all the charts KNN, RF, RTC, and MEWMA-SS do not have a reliable IC performance in Case V when the actual serial data correlation is non-stationary. Fourth, the IC performance of KC is poor in all cases considered here. From Table S.1 in the supplementary file, it can be seen that values of the standard deviation of ARL_0, denoted as $SDARL_0$, of the charts TSL-S and TSL-NS are generally smaller than or comparable to those of the charts KNN, RF, and RTC, except in cases when the estimated ARL_0 values of the latter three charts are well below the nominal ARL_0 value of 200 (e.g., in cases when $m_0 \leq 300$). The $SDARL_0$ values of KC are small in all cases due to its small estimated ARL_0 values. The $SDARL_0$ values of MEWMA-SS are comparable to those of TSL-S and TSL-NS, except in Case V when its estimated ARL_0 values are too small. Generally speaking, control charts based on conventional machine leaning techniques require a large training data to have a reasonably reliable classification rule since the specific process characteristics to learn are usually not well specified in advance. This is true especially when the process observations have complicated data structure. This explains why the four competing machine learning charts KC, KNN, RF and RTC do not perform well in some cases considered here. As a comparison, the TSL-based monitoring charts MEWMA-SS, TSL-S and TSL-NS have the process characteristics to learn well specified in advance so that their learning process could be more efficient. Furthermore, the IC dataset keeps expanding in these TSL-based charts if the process is declared to be IC during online process monitoring, making the learned process characteristics more accurate in estimating the true IC process characteristics and the related

charts more effective. The TSL-based charts MEWMA-SS and TSL-S do not perform well in some cases because their model assumptions (e.g., normality in MEWMA-SS and stationarity in TSL-S) are violated in such cases. So, we suggest using TSL-NS in cases when these assumptions cannot be confirmed. The estimated ARL_0 and $SDARL_0$ values of the charts KC, KNN, RF, RTC, and MEWMA when they are applied to the original process observations without data decorrelation in Cases I-V are also computed, and presented in Table S.2 of the supplementary file. It can be seen from the table that their IC performance is generally unreliable in cases when serial data correlation exists but data de-correlation is not implemented properly in advance.

The results in Figure 2 are for cases when the number of quality characteristics p is 5. Intuitively, when p is larger, the necessary IC sample size m_0 should be larger as well to have a reliable IC performance of a TSL-based control chart. To confirm this, we first consider an extension of Case IV to a p-dimensional case where observations of each quality characteristic are generated in the same way as that in Case IV. Because the serial data correlation is stationary in such a case, we focus on the IC performance of the chart TSL-S. When p changes among 1, 3, 5, 7, and 10 and other setups are the same as those in Figure 2, the estimated ARL_0 and $SDARL_0$ values of TSL-S are presented in Table 1. From the table, it can be seen that the necessary IC sample size m_0 indeed should be larger to have a reliable IC performance of TSL-S when p is larger. In a case extended from Case V when the serial data correlation is nonstationary, the estimated ARL_0 and $SDARL_0$ values of TSL-NS when p changes among 1, 3, 5, 7, and 10 are presented in Table S.3 of the supplementary file. A similar conclusion can be made from that table.

Evaluation of the OC performance: Next, we evaluate the OC performance of the related charts in cases when $m_0 = 500$, all quality characteristics have a same shift at the beginning of online process monitoring with the size varying among ± 0.25, ± 0.5, ± 0.75 or ± 1.0, and other setups are the same as those in Figure 2. To make the comparison among different charts meaningful, their control limits have been adjusted properly so that their actual ARL_0 values all equal the nominal level of 200. First, we consider cases when the related parameters of the charts are chosen to be the same as those in the example of Figure 2. The results of the computed ARL_1 values of the five charts in Cases I–V are presented in Figure S.1 of the supplementary file. From the figure, we can have the following conclusions. (i) The chart TSL-S performs the best or close to the best among all five charts in Cases I–IV when the serial data correlation is stationary, especially when the shift size is relatively small. (ii) The chart TSL-NS performs the best in most scenarios in Case

Table 1. Estimated ARL_0 values and $SDARL_0$ values (in parentheses) of TSL-S in a case extended from Case IV when the nominal ARL_0 value is fixed at 200 and p changes among 1, 3, 5, 7, and 10.

p	$m_0 = 200$	300	400	500	1000	2000
1	190 (46.6)	193 (44.8)	195 (42.7)	196 (32.7)	198 (21.9)	197 (14.9)
3	183 (45.4)	192 (36.5)	191 (27.8)	195 (23.9)	198 (18.1)	198 (12.2)
5	158 (48.0)	189 (30.5)	193 (28.5)	195 (28.2)	199 (23.7)	200 (20.0)
7	125 (30.6)	144 (37.5)	153 (39.8)	176 (40.1)	194 (32.9)	196 (30.2)
10	93 (32.1)	110 (34.4)	126 (40.5)	147 (48.7)	160 (54.2)	181 (50.3)
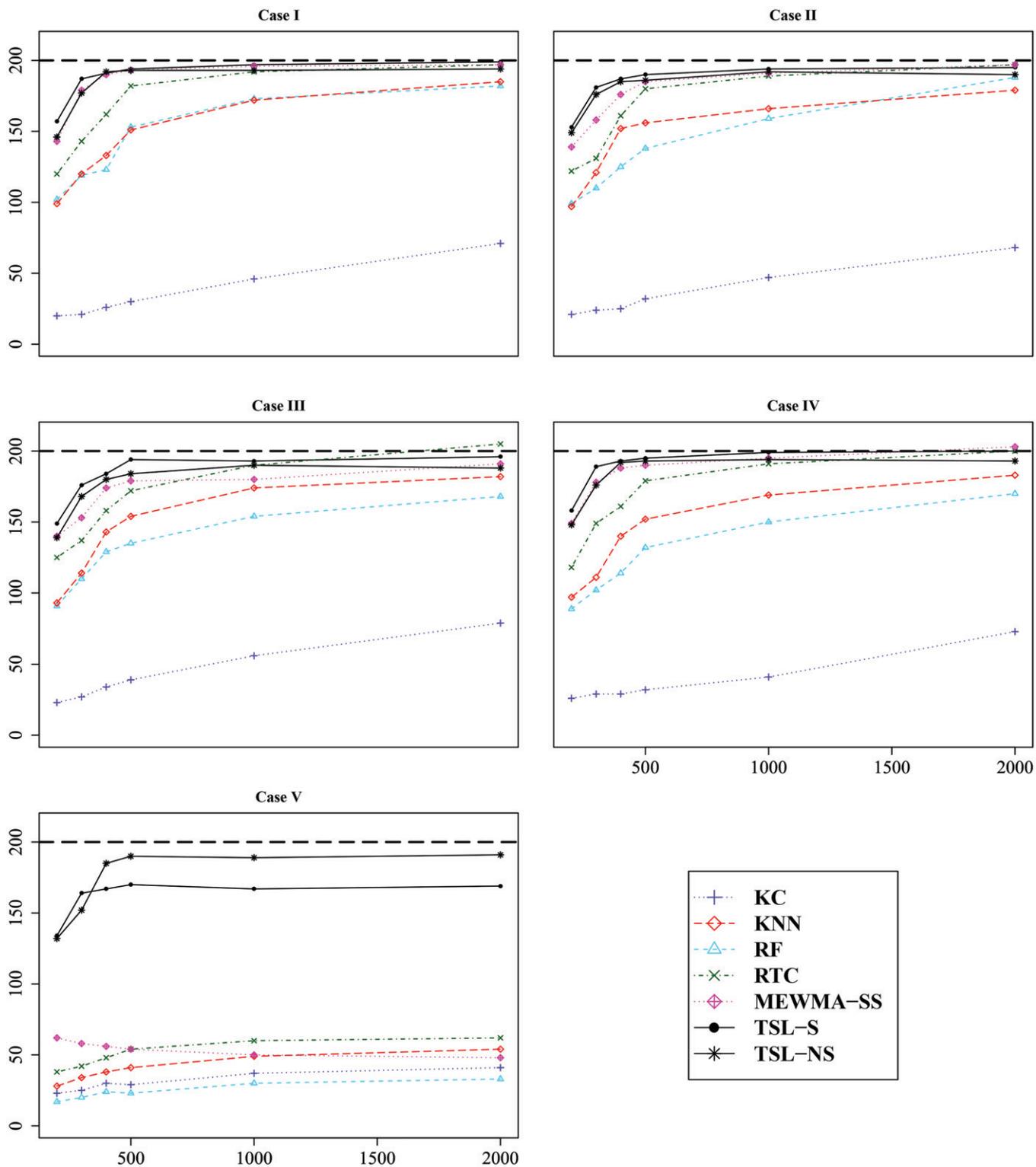
Figure 2. Estimated ARL_0 values of seven control charts applied to the decorrelated data when their nominal ARL_0 values are fixed at 200, and the IC sample size m_0 changes among 200, 300, 400, 500, 1000, and 2000.

V when the serial data correlation is non-stationary, and its performance is reasonable in Cases I–IV, although it is less effective than the chart TSL-S in such cases. (iii) The chart MEWMA-SS performs reasonably well in Cases I, II and IV when the normality assumption is valid, and it is less effective in Cases III and V when the normality assumption is invalid, compared to TSL-S in Case III and TSL-NS in Case V. (iv) The charts KC,

KNN and RTC have a reasonably good performance in cases when the shift size is relatively large (e.g., ±0.75 and ±1.0), but the chart RF does not have a satisfactory performance in all cases considered. (v) Among the four alternative charts KC, KNN, RF, and RTC, it seems that RTC has the best OC performance. The above conclusions (i)–(iii) are intuitively reasonable. The reason why the alternative charts KC, KNN, and RTC have a

reasonably good performance when the shift size is relatively large can be explained intuitively as follows. Remember that the decision rules of the charts KC and KNN are determined from the IC dataset obtained before online process monitoring and then the decision at a given time point during online process monitoring is made based on the observed data at that time point only. So, these two charts are basically Shewhart charts. The RTC chart is different from them in that it determines its decision rule based on the IC dataset and the observed data in a moving window of the current time point. Its decision at the current time point is then made based on the observed data at the current time point. Thus, it is also a Shewhart chart although its decision rule is determined adaptively during online process monitoring. Generally speaking, Shewhart charts are effective in detecting large shifts (Qiu 2014, chap. 3), which is confirmed in this example. Also, by comparing the performance of the charts RF and RTC in this example, we can see that the adaptive way to determine the decision rule in RTC did improve the performance of an artificial contrast chart like RF. Some results when the charts KC, KNN, RTC and MEWMA are applied to the original process observations without considering data decorrelation are presented in Figures S.2 and S.3 of the supplementary file. Similar conclusions to those from Figure S.1 can be made from these results.

In the example of Figure S.1 in the supplementary file, the related parameters in the seven charts are chosen as in the example of Figure 2. In such cases, the OC performance of the charts may not be comparable, as pointed out in the literature (e.g., Qiu 2018). To avoid this limitation, in the following example, we compare the optimal OC performance of all seven charts by choosing the parameters of each chart such that its ARL_1 value reaches the minimum for detecting a given shift. All other setups are kept to be the same as those in the previous example. The optimal ARL_1 values of the seven control charts in cases considered in Figure S.1 are shown in Figure 3. From the figure, it can be seen that similar conclusions to those in the previous example can be made here regarding the OC performance of the seven charts. Some results when the charts KC, KNN, RTC and MEWMA are applied to the original process observations without considering data de-correlation are presented in Figures S.4 and S.5 of the supplementary file. Similar conclusions to those from Figure 3 can be made from these figures.

To study the impact of the initial IC data size m_0, in the setup of Figure S.1, the calculated ARL_1 values of the charts TSL-S and TSL-NS when m_0 changes among 200, 300, 400, 500, 1000, and 2000 are presented in Figures S.6 and S.7 of the supplementary file, respectively. From the figures, it can be seen that the OC performance of TSL-S and TSL-NS is generally better when m_0 is larger, and their OC performance is quite stable when $m_0 \geq 500$ in all cases considered.

4. An Application

We demonstrate the application of the seven charts KC, KNN, RF, RTC, MEWMA-SS, TSL-S, and TSL-NS considered in the previous section by using a dataset about a semiconductor manufacturing process that is maintained by the UC Irvine Machine Learning Repository (*http://archive.ics.uci.edu/ml/*

datasets/SECOM). This dataset contains observations of many extracted quality characteristics of the semiconductor products sampled from the manufacturing process, and has become a standard dataset for testing various multivariate control charts in the SPC literature (e.g., Chen, Zi, and Zou 2016; Li et al. 2020). For a general discussion about statistical monitoring of semiconductor manufacturing processes, see Yashchin (2018). Here, we choose five quality characteristics to test the related charts. The original data of these five quality features are shown in Figure S.5 in the supplementary file. From the plots in the figure, it can be seen that the first 500 observations of the data are quite stable over time. Thus, they are used as the IC data. For these IC data, we first check the significance of the autocorrelation for all quality characteristics using the Durbin–Watson test. The p-values of this test for the five quality characteristics are 1.789×10^{-3}, 4.727×10^{-1}, 4.760×10^{-4}, 1.412×10^{-4}, and 9.744×10^{-2}. So, there is significant autocorrelation in the time series of the first, third and fourth quality characteristics. To check the stationarity of the autocorrelation, the augmented Dickey–Fuller (ADF) test is used and the p-values of this test for five quality characteristics are all < 0.01, implying that the stationary assumption is valid in this example. To check the correlation among the five quality characteristics in the IC data, the following sample correlation coefficient matrix is first computed:

$$
\begin{bmatrix}
1.000 & -0.046 & 0.106 & 0.054 & 0.012 \\
-0.046 & 1.000 & -0.131 & -0.036 & -0.104 \\
0.106 & -0.131 & 1.000 & 0.413 & -0.059 \\
0.054 & -0.036 & 0.413 & 1.000 & 0.393 \\
0.012 & -0.104 & -0.059 & 0.393 & 1.000
\end{bmatrix}.
$$

Then, the Pearson's correlation test is used to check pairwise correlation among the five quality characteristics, and the following pairs are found to be significantly correlated (X_1, X_3), (X_2, X_3), (X_2, X_5), (X_3, X_4), and (X_4, X_5). The corresponding p-values are respectively 0.018, 0.003, 0.020, 5.259×10^{-22}, and 7.088×10^{-20}. Therefore, based on the above data analysis, the IC data have significant stationary serial data correlation and significant correlation among the five quality characteristics.

We then decorrelate the data using the data decorrelation procedure for stationary correlated data discussed in Section 2.2, and the decorrelated data are shown in Figure S.9 of the supplementary file. After the data decorrelation, the p-values of the Durbin-Watson test for checking the autocorrelation of the five quality characteristics in the decorrelated IC data are 0.444, 0.495, 0.489, 0.489, and 0.483, respectively. Thus, there is no significant evidence of autocorrelation in the decorrelated IC data. All pairwise Pearson's correlation coefficients computed from the decorrelated IC data are < 0.0015, implying the pairwise correlation among the five quality characteristics has been mostly removed as well. To check the normality assumption of the decorrelated data, the Shapiro test is used, and its p-values for the five quality characteristics are 4.991×10^{-9}, 3.573×10^{-17}, 3.481×10^{-9}, 0.012, and 1.113×10^{-8}, respectively. Thus, the normality assumption is significantly violated for all five quality characteristics.

Next, we apply the seven charts KC, KNN, RF, RTC, MEWMA-SS, TSL-S, and TSL-NS to the decorrelated data for online process monitoring, starting from the 501st observation.
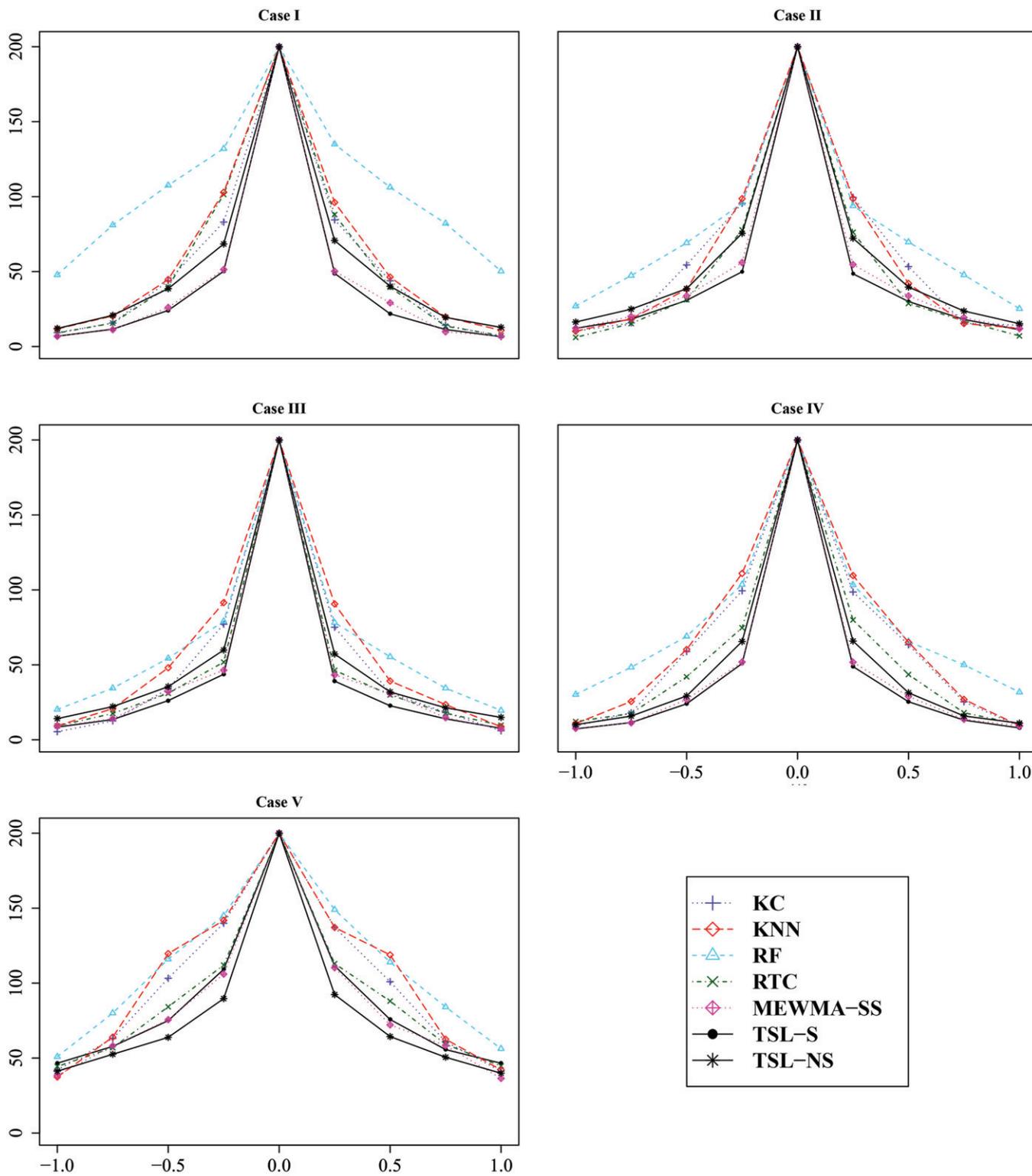
Figure 3. Optimal ARL$_1$ values of the seven control charts when their nominal ARL$_0$ values are fixed at 200, $p = 5$, $m_0 = 500$, all quality characteristics have the same shift, and the shift size changes among ± 0.25, ± 0.5, ± 0.75, and ± 1.0.

The control limits of these charts are computed as in Section 3. The five charts are shown in Figure 4. From the charts, it can be seen that the first signals of the charts KC, KNN, RF, RTC, MEWMA-SS, TSL-S, and TSL-NS are at the time points 537, 534, 543, 529, 523, 521, and 532, respectively. Thus, the chart TSL-S gives the earliest signal in this example. To verify the

detected shift, a multivariate change-point detection procedure (cf., Qiu 2014, sec. 7.5) is applied to the decorrelated Phase II data with the observation times in the range [501,600], and the detected change-point is 517. Then, the Hotelling's T^2 test to compare the multivariate means of the two groups of data with the observation times in [501,516] and [517,600] gives
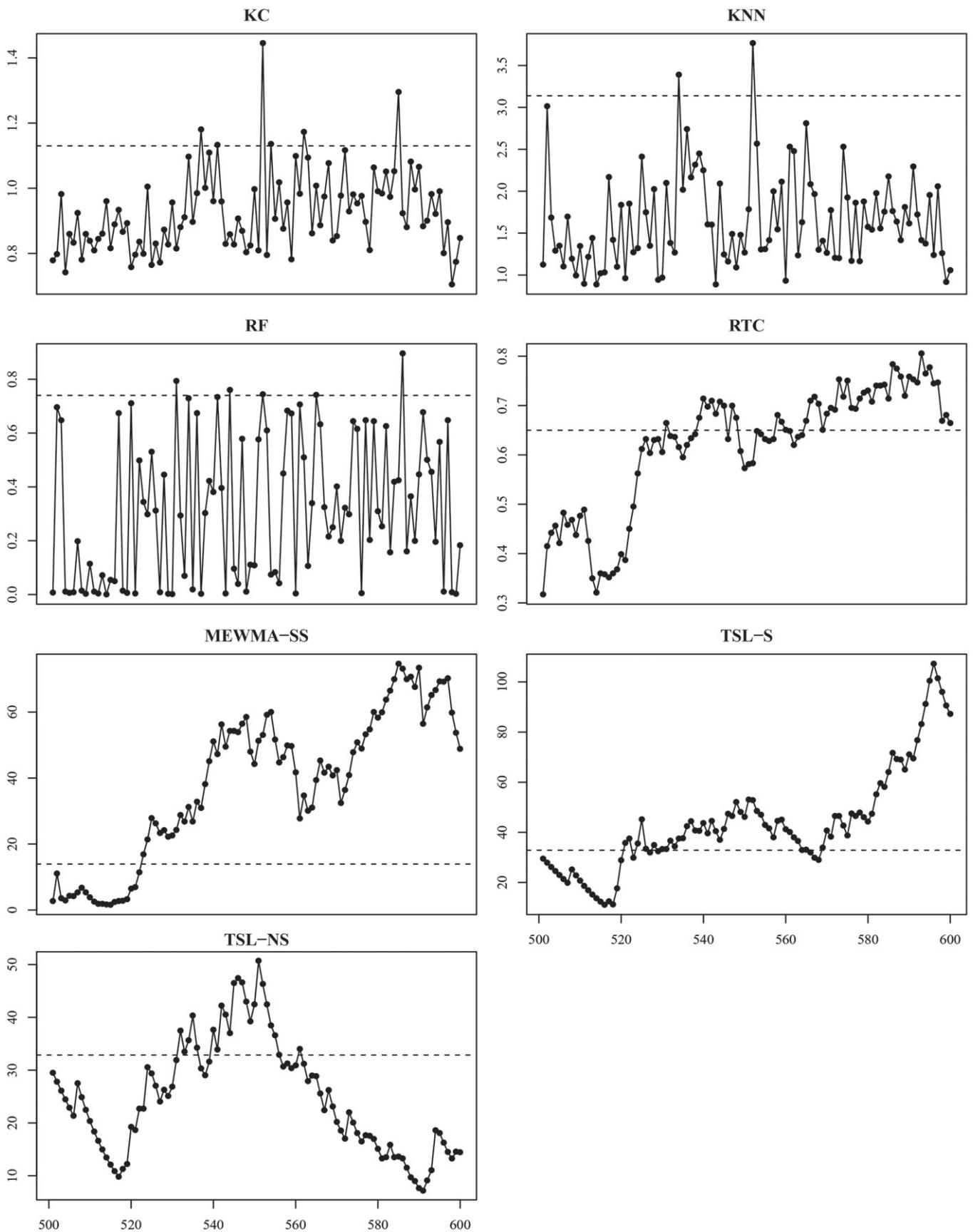
Figure 4. Control charts KC, KNN, RF, RTC, MEWMA-SS, TSL-S, and TSL-NS for monitoring the decorrelated process observations in the semiconductor manufacturing example.

the p-value of 4.426×10^{-3}. Thus, the multivariate process mean has a significant shift at time 517. To identify the specific quality characteristics that have mean shifts at time 517, the t-test to compare the means of individual quality characteristics in the two groups of data with the observation times in [501,516] and [517,600] gives the p-values of 1.581×10^{-3}, 0.168, 0.022, 0.726, and 0.092, respectively, for the five quality characteristics. Thus, it seems that the first and third quality characteristics have significant mean shifts and the fifth quality characteristic has a marginally significant mean shift, which is quite consistent with the plots in Figure S.9 of the supplementary file.

From Figure S.9 of the supplementary file, there seems an obvious outlier in X_2^* at time 299 and another outlier in X_5^* at time 169. Because the number of outliers is small relative to the IC sample size, we expect that they would not have a substantial impact on the results. To confirm this, we re-run the above analyses, after the observations at times 169 and 299 are removed from the IC data. Indeed, the results are not changed much, and the first signals of the charts KC, KNN, RF, RTC, MEWMA-SS, TSL-S and TSL-NS are at the time points 537, 534, 531, 531, 523, 521, and 532, respectively, in such a case.

5. Concluding Remarks

In recent years, some machine leaning methods have been suggested for process monitoring and control in the SPC literature, using artificial contrast, real-time contrast, one-class-classification, and some other ideas. In the SPC research and practice, certain IC parameters are routinely estimated from an IC dataset before online process monitoring. To make the estimates more accurate and accommodate situations when the IC data size is relatively small, self-starting charts have become a popular choice for online process monitoring. Based on that idea, we propose the new learning framework TSL in this article, in which process characteristics to learn from the observed data are well specified in advance. Online process monitoring based on TSL consists of several steps that are demonstrated in Figure 1, which can accommodate serial data correlation and nonparametric IC process distribution. It has been shown numerically that the TSL-based control charts are more reliable and effective in various cases, compared to some existing methods based on machine learning algorithms. In the current article, the TSL-based online process monitoring schemes are demonstrated for monitoring a multivariate process when the serial data correlation is short-range and the IC process distribution can be described well by its mean and covariance matrix. Although these assumptions are already quite flexible and can provide a reasonable approximation to some SPC applications, some real applications could be excluded by them. So, future research is needed to extend the TSL-based online process monitoring schemes discussed in this article to certain more general scenarios. In addition, from Table 1 in Section 3 and Table S.3 in the supplementary file, when the number of quality characteristics p is larger, the required IC data size m_0 should be larger to have a reliable TSL-based online process monitoring. In some applications, however, a large IC dataset may not be available. One possible solution is to include a dimension reduction or variable selection procedure in the proposed TSL framework to reduce the dimensionality or the number of IC parameters that need to be estimated from the IC data. This issue will be addressed in our future research.

Supplementary Materials

ComputerCodesAndData.zip: This zip file contains Fortran source codes of our proposed method and the real data used in the article.
supplement.pdf: This supplementary file contains (i) a description of the nonparametric CUSUM chart by Qiu (2008), and (ii) some extra numerical results.

Acknowledgments

References

Aggarwal, C. C. (2018), *Neural Networks and Deep Learning*, New York: Springer. [487]

Altmann, E. G., Cristadoro, G., and Esposti, M. D. (2012), "On the Origin of Long-Range Correlations in Texts," *Proceedings of the National Academy of Sciences of the USA*, 109, 11582–11587. [493]

Capizzi, G., and Masarotto, G. (2008), "Practical Design of Generalized Likelihood Ratio Control Charts for Autocorrelated Data," *Technometrics*, 50, 357–370. [487]

Capizzi, G., and Masarotto, G. (2013), "Phase I Distribution-Free Analysis of Univariate Data," *Journal of Quality Technology*, 45, 273–284. [489]

Chakraborti, S., and Graham, M. A. (2019), *Nonparametric Statistical Process Control*, New York: Wiley. [487]

Chatterjee, S., and Qiu, P. (2009), "Distribution-Free Cumulative Sum Control Charts Using Bootstrap-Based Control Limits," *Annals of Applied Statistics*, 3, 349–369. [490]

Chen, N., Zi, X., and Zou, C. (2016), "A Distribution-Free Multivariate Control Chart," *Technometrics*, 58, 448–459. [497]

Cressie, N. (1990), "The Origins of Kriging", *Mathematical Geology*, 22, 239–252. [490]

Crosier, R B. (1998), "Multivariate Generalizations of Cumulative Sum Quality-Control Schemes," *Technometrics*, 30, 291–303. [487]

Deng, H., Runger, G., and Tuv, E. (2012), "System Monitoring With Real-Time Contrasts," *Journal of Quality Technology*, 44, 9–27. [488,493,494]

Epanechnikov, V. A. (1969), "Non-Parametric Estimation of a Multivariate Probability Density," *Theory of Probability and Its Applications*, 14, 153–158. [490]

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning – Data Mining, Inference, and Prediction,* Berlin: Springer-Verlag. [487]

Hawkins, D. M. (1987), "Self-Starting Cusums for Location and Scale," *The Statistician*, 36, 299–315. [488]

Hawkins, D. M., and Maboudou-Tchao, E. M. (2007), "Self-Starting Multivariate Exponentially Weighted Moving Average Control Charting," *Technometrics*, 49, 199–209. [488]

Hawkins, D. M., and Olwell, D. H. (1998), *Cumulative Sum Charts and Charting for Quality Improvement*, New York: Springer. [487]

Hawkins, D. M., Qiu, P., and Kang, C. W. (2003), "The Changepoint Model for Statistical Process Control," *Journal of Quality Technology*, 35, 355–366. [487]

He, S., Jiang, W., and Deng, H. (2018), "A Distance-Based Control Chart for Monitoring Multivariate Processes Using Support Vector Machines," *Annals of Operations Research*, 263, 191–207. [488]

Hu, J., Runger, G., and Tuv, E. (2007), "Tuned Artificial Contrasts to Detect Signals," *International Journal of Production Research*, 45, 5527–5534. [488]

Hwang, W., Runger, G., and Tuv, E. (2007), "Multivariate Statistical Process Control With Artificial Contrasts," *IIE Transactions*, 2, 659–669. [488,493]

Jones-Farmer, L. A., Woodall W. H., Steiner, S. H., and Champ, C. W. (2014), "An Overview of Phase I Analysis for Process Improvement and Monitoring," *Journal of Quality Technology*, 46, 265–280. [488,489]

Keefe, M. J., Woodall, W. H., and Jones-Farmer, L. A. (2015), "The Conditional In-Control Performance of Self-Starting Control Charts," *Quality Engineering*, 27, 488–499. [488]

Li, F., Runger, G. C., and Tuv, E. (2006), "Supervised Learning for Change-Point Detection," *International Journal of Production Research*, 44, 2853–2868. [488]

Li, W., Xiang, D., Tsung, F., and Pu, X. (2020), "A Diagnostic Procedure for High-Dimensional Data Streams Via Missed Discovery Rate Control," *Technometrics*, 62, 84–100. [497]

Lowry, C. A., Woodall, W. H., Champ, C. W., and Rigdon, S. E. (1992), "Multivariate Exponentially Weighted Moving Average Control Chart," *Technometrics*, 34, 46–53. [487,492]

Megahed, F.M., and Jones-Farmer, L.A. (2015), "Statistical Perspectives on 'Big Data,'" In *Frontiers in Statistical Quality Control* (Vol. 11), eds. S. Knoth and W. Schmid, Cham: Springer, 29–47. [488]

Montgomery, D. C. (2012), *Introduction to Statistical Quality Control*, New York: Wiley. [487]

Prajapati, D. R., and Singh, S. (2012), "Control Charts for Monitoring the Autocorrelated Process Parameters: A Literature Review," *International Journal of Productivity and Quality Management*, 10, 207–249. [487]

Qiu, P. (2008), "Distribution-Free Multivariate Process Control Based on Log-Linear Modeling," *IIE Transactions*, 40, 664–677. [492,500]

—— (2014), *Introduction to Statistical Process Control*, Boca Raton, FL: Chapman Hall/CRC. [487,488,489,497,498]

—— (2018), "Some Perspectives on Nonparametric Statistical Process Control," *Journal of Quality Technology*, 50, 49–65. [487,492,497]

—— (2020), "Big Data? Statistical Process Control Can Help!" *The American Statistician*, 74, 329–344. [487]

Reis, M.S., and Gins, G. (2017), "Industrial Process Monitoring in the Big Data/Industry 4.0 Era: From Detection, to Diagnosis, to Prognosis," *Processes*, 5, 35. [487]

Sall, J. (2018), "Scaling-up Process Characterization," *Quality Engineering*, 30, 62–78. [489]

Sukchotrat, T., Kim, S. B., and Tsung, F. (2010), "One-Class Classification-Based Control Charts for Multivariate Process Monitoring," *IIE Transactions*, 42, 107–120. [488,493,494]

Sun, R., and Tsung, F. (2003), "A Kernel-Distance-Based Multivariate Control Chart Using Support Vector Methods," *International Journal of Production Research*, 41, 2975–2989. [488,493]

Tax, D. M., and Duin, R. P. W. (2004), "Support Vector Data Description," *Machine Learning*, 54, 45–66. [488]

Tuv, E., and Runger, G. (2003), "Learning Patterns Through Artificial Contrasts With Application to Process Control," *Transactions on Information and Communications Technologies*, 29, 63–72. [488]

Weese, M., Martinez, W., Megahed, F. M., and Jones-Farmer, L. A. (2016), "Statistical Learning Methods Applied to Process Monitoring: An Overview and Perspective," *Journal of Quality Technology*, 48, 4–24. [488]

Xue, L., and Qiu, P. (2020), "A Nonparametric CUSUM Chart for Monitoring Multivariate Serially Correlated Processes," *Journal of Quality Technology*, DOI: 10.1080/00224065.2020.1778430. [487]

Yang, K., and Qiu, P. (2020), "Online Sequential Monitoring of Spatio-Temporal Disease Incidence Rates," *IISE Transactions*, 52, 1218–1233. [487]

Yashchin, E. (2018), "Statistical Monitoring of Multi-Stage Processes," in *Frontiers in Statistical Quality Control*, eds. S. Knoth and W. Schmid, Vol. 12, 185–209, Berlin: Springer. [497]

You, L., and Qiu, P. (2019), "Fast Computing for Dynamic Screening Systems When Analyzing Correlated Data," *Journal of Statistical Computation and Simulation*, 89, 379–394. [490]

Zhang, C., Tsung, F., and Zou, C. (2015), "A General Framework for Monitoring Complex Processes With Both In-Control and Out-of-Control Information," *Computers & Industrial Engineering*, 85, 157–168. [488]

Zou, C., Zhou, C., and Wang, Z. (2007), "A Self-Starting Control Chart for Linear Profiles," *Journal of Quality Technology*, 39, 364–375. [488]