Analytica Chimica Acta 908 (2016) 75–84

Contents lists available at ScienceDirect

Analytica Chimica Acta

journal homepage: www.elsevier.com/locate/aca

# Confidence limits for contribution plots in multivariate statistical process control using bootstrap estimates

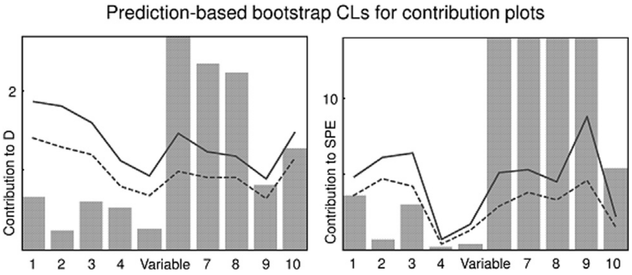
Hamid Babamoradi*, Frans van den Berg, Åsmund Rinnan

*University of Copenhagen, Faculty of Science, Department of Food Science, Spectroscopy & Chemometrics Section, Rolighedsvej 30, DK-1958 Frederiksberg, Denmark*

## HIGHLIGHTS

- Bootstrap can offer CLs for contribution plots, unlike asymptotic methods.
- New bootstrap CLs are suggested for contribution plots in PCA-based batch MSPC.
- CLs for contribution plots can help in fault diagnosis.
- Performance of bootstrap CLs for MSPC control charts were shown before.
- Bootstrap CLs for contribution plots were compared to the CLs based on existing methods.

## GRAPHICAL ABSTRACT



## ARTICLE INFO

*Article history:*
Received 18 September 2015
Received in revised form
4 January 2016
Accepted 5 January 2016
Available online 7 January 2016

*Keywords:*
Variable contributions
Multivariate statistical process control
Bootstrap
Process monitoring
Confidence limit
Fault diagnosis

## ABSTRACT

In Multivariate Statistical Process Control, when a fault is expected or detected in the process, contribution plots are essential for operators and optimization engineers in identifying those process variables that were affected by or might be the cause of the fault. The traditional way of interpreting a contribution plot is to examine the largest contributing process variables as the most probable faulty ones. This might result in false readings purely due to the differences in natural variation, measurement uncertainties, etc. It is more reasonable to compare variable contributions for new process runs with historical results achieved under Normal Operating Conditions, where confidence limits for contribution plots estimated from training data are used to judge new production runs. Asymptotic methods cannot provide confidence limits for contribution plots, leaving re-sampling methods as the only option. We suggest bootstrap re-sampling to build confidence limits for all contribution plots in online PCA-based MSPC. The new strategy to estimate CLs is compared to the previously reported CLs for contribution plots. An industrial batch process dataset was used to illustrate the concepts.

## 1. Introduction

Multivariate Statistical Process Control (MSPC) [1] is widely used as a process analytical chemistry and technology tool in industries like food, chemical, pharmaceutical, and petroleum manufacturing. As a general concept, one unit operation or a complete production system is monitored over time through measurements of a number of process variables with the purpose of checking if the system is in statistical control. This is achieved by projecting the collected data into multivariate control charts. The

* Corresponding author.
*E-mail address:* hamba@food.ku.dk (H. Babamoradi).

http://dx.doi.org/10.1016/j.aca.2016.01.002
0003-2670/© 2016 Elsevier B.V. All rights reserved.

control charts plus control/confidence limits (CLs) are built based on previous recordings under Normal Operating Conditions (NOC). After training, the charts can be used to detect special events and possibly identify the corresponding causes of these events. The aim is to continuously improve the process and thus product consistency and quality. Once a production run exceeds the limits in the control chart a fault can be reported to process operators. Contribution plots are then examined to identify those process variables which were potentially affected by and/or causing the fault [2,3]. Several studies have shown that using CLs for contribution plots is very helpful in identifying the process variable(s) affected by the fault [4,5], stressing that the relative size of contributions, compared to the NOC contributions, should be examined instead of simply the absolute size of the contributions.

Traditional CLs in control charts are based on asymptotic methods that are used under the assumption of a predefined distribution - e.g. the normal-, F-, or chi-squared-distributions - for the uncertainties in the model and residuals [1,5]. There are no CLs for variable contributions based on asymptotic methods [4,5]. Bootstrap re-sampling [6–8], on the other hand, offers CLs for all model parameters and measures. They have shown to be compatible with the complexities of manufacturing processes, since re-sampling methods can take “impulsive” process factors into account by estimating uncertainty distributions directly from data under NOC [9]. In addition, they can provide CLs for contributions unlike the presently applied asymptotic methods [4].

Several papers have been published on bootstrap confidence estimation for control charts in MSPC [9–14], showing the potential of re-sampling methods compared to asymptotic methods. Nonparametric bootstrapping has previously been used to estimate standard deviations of model loadings to build CLs for contributions of variables to principal component scores by Conlin et al. (2000) [4]. The approach was compared on covariance and correlation data of a simulated continuous process. The theory of contribution plots was extended to latent variable models with correlated scores, and CLs for the contribution plots were introduced by Westerhuis et al. (2000) [5]. CLs of the contributions plots for the D-statistic were based on jackknifing, while for the Q-statistic CLs based on asymptotic methods were suggested. No previous studies have used the bootstrap re-sampling method to estimate the CL for all the contribution plots. Furthermore, the suggested methods to calculate CLs for contributions are based on asymptotic theory where only the standard error of the contributions were estimated using re-sampling methods such as jackknifing [5] and bootstrapping [4]. In this study we present nonparametric bootstrap CLs for all the contribution plots in online Principal Component Analysis (PCA) based MSPC, building on methods previously introduced to determine reliable bootstrap-based CLs for control charts [9]. The bootstrap CLs were compared with the CLs introduced by Westerhuis et al. (2000) [5] for both Q- (based on asymptotic approximation) and D-statistic (based on jackknifing). A well-known industrial dataset on the polymerization of Nylon was used to show the performance of the variable contribution CLs [1,15].

## 2. PCA-based MSPC

We refer to the work by Nomikos et al. [1] and Westerhuis et al. [5] for the full details on how to build control charts and contribution plots in PCA-based MSPC. A synchronized batch dataset forms a natural three-way tensor $\underline{\mathbf{X}}(I \times J \times K)$, where $I$ represents number of batches (in the training set), $J$ the number of process variables measured, and $K$ the number of time intervals. In this paper we consider $\mathbf{X}$ as NOC batch data that are unfolded to a two-way array $\mathbf{X}(I \times JK)$. In this procedure the batch mode stays intact, whereas the variable and time modes are nested. The data are auto-scaled to have unit variance and average zero for each column of $\mathbf{X}$. In PCA-based MSPC the starting point is:

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^{\mathrm{T}} + \mathbf{t}_2\mathbf{p}_2^{\mathrm{T}} + \ldots + \mathbf{t}_R\mathbf{p}_R^{\mathrm{T}} + \mathbf{E} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{1}$$

$\mathbf{T}$ stands for scores, $\mathbf{P}$ for loadings, and $\mathbf{E}$ for the residuals. $R$ is the number of Principal Components (PCs) retained in the modeled part. Once the model is built, a new batch can be evaluated as follows:

$$\mathbf{t}_{\mathrm{new}} = \mathbf{x}_{\mathrm{new}}\mathbf{P} \tag{2}$$

$$\mathbf{e}_{\mathrm{new}} = \mathbf{x}_{\mathrm{new}} - \mathbf{t}_{\mathrm{new}}\mathbf{P}^{\mathrm{T}} \tag{3}$$

Note that the mean and standard deviation values that are used to center and scale $\mathbf{X}$ are also applied to center and scale the new batch data record before projection.

### 2.1. Batch MSPC strategies

Two out of four commonly applied PCA-based MSPC strategies, so-called *local* and *evolving models*, are included in this manuscript to illustrate the concept of bootstrap CLs for process variable contributions, as they are simple to understand; the procedure is directly applicable to the other strategies [9].

A local model method takes only the variations for each time point $k$ into account [16]. A PCA model is fitted to data for each time point, $\mathbf{X}_k(I \times J)$, which leads to in total $K$ (separate) models. The statistics are also calculated for each time point separately, meaning that residuals of each model are used to calculate SPE statistic while PC scores are used to calculate D-statistic for that specific time point (see section 2.2. for details on how to calculate each statistic). Data of new batches for each time point are projected onto the loadings for that corresponding interval, $\mathbf{P}_k(R \times J)$, and then the monitoring statistics are determined. Note that the local model method is technically neither a batch-wise nor a variable-wise approach as no unfolding is used - the data of each time point are analyzed independently.

An evolving model method takes the history of the batch into considerations while the process evolves [16]. A model is fitted to data up to each time point $k$, $\mathbf{X}_k(I \times Jk)$, which adds up to a total of $K$ models. The first model only accounts for process variations in the first time point (equal to the local model), while the $K^{\mathrm{th}}$ model accounts for the whole process variation. The statistics are, however, calculated for each time point separately. Data of new batches up to each time point are projected onto the loadings for that interval, $\mathbf{P}_k(R \times Jk)$, and then the statistics are calculated and compared with the control limits. To distinguish between local and evolving model it is important to emphasize ones more that in local model we fit a PCA model to the data of each time point $\mathbf{X}_k(I \times J)$, while in evolving model we fit a PCA model to the data from the first time point up to the current time point $\mathbf{X}_k(I \times Jk)$.

### 2.2. Control charts

Control charts are used to monitor each batch and detect special events or Abnormal Operating Conditions (AOC) for the process. There are three standard multivariate control charts employed to evaluate and assess the performance of a new batch in manufacturing: SPE, D-statistic, and score(s).

SPE (Squared Prediction Error) is the distance of a batch from the model space at each time point. SPE is also called Q-statistic [1,17], however we use SPE throughout the manuscript to avoid confusion. It captures how well/poor the data are explained by the model for the current time points by simply summing variations unexplained by the model:

$$SPE_{ik} = \sum_{j=1}^{J} e_{i(j)k}^2 \tag{4}$$

$SPE_{ik}$ is the value of SPE for batch $i$ in the current time point $k$, and $J$ is the number of variables. Low values of SPE for a batch show that the variations in the batch are explained well by the PCA model, whereas the opposite is true for the high values.

The most common way to estimate CLs for the SPE is to assume a normal distribution, based on the work by Jackson and Mudholkar [18]:

$$SPE_\alpha \sim \theta_1 \left[ \frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} + 1 \right]^{1/h_0} \tag{5}$$

where the parameters (moments) of the approximation are defined as

$$\theta_l = \sum_{i=R+1}^{I} \lambda_{(i)}^l \quad (l = 1, 2, 3) \tag{6}$$

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{7}$$

$I$ is the number of objects/batches used in training, $R$ is number of PCs, $\lambda$ is an eigen value of the covariance or correlation matrix of the NOC data, and $c\alpha$ is the normal distribution at the $\alpha$ significance level. The Jackson and Mudholkar approach is perceived more robust when the training/NOC sample size is relatively small or in the presence of outliers [1].

D-statistic (or Hotelling's $T^2$) is the distance of a batch from the center of the model (the Mahalanobis distance in the projection subspace), which represents variations explained by the model for this object/batch:

$$D_i = \mathbf{t}_i^T \mathbf{S}^{-1} \mathbf{t}_i \tag{8}$$

$D_i$ is the value of the $D$-statistic for batch $i$, $\mathbf{t}_i$ is the score vector for batch $i$, and $\mathbf{S}^{-1}$ is the inverse of the covariance matrix (correlation matrix in case of auto-scaled data) calculated from the score matrix for NOC data. CLs for the D-statistic from NOC batches - the training set - can be approximated using the $F$-distribution:

$$D_\alpha \sim \frac{(I-1)^2}{I} \times \frac{(R/(I-R-1))F(R, I-R-1, \alpha)}{1 + (R/(I-R-1))F(R, I-R-1, \alpha)} \tag{9}$$

while the CL for new batches is most frequently estimated by Ref. [19]:

$$D_\alpha \sim \frac{R(I^2 - 1)}{I(I-R)} F(R, I-R, \alpha) \tag{10}$$

where $I$ is the number NOC batches, $R$ stands for number of PCs retained in the model, and $F$ the Fisher's $F$-distribution at the $\alpha$ significance level.

Where D-statistics monitors the overall variation inside the model, score control charts show the distance of a batch from the center of the model along each (orthogonal) PC separately. There can be AOC situations where only one of the charts – D or one of the scores - will detect an out-of-control behavior. Therefore having both versions might help to improve overall monitoring performance. Score values are estimated from Equation (1) for NOC batches and from Equation (2) for a new batch. Upper and lower CLs for each score chart are estimated by Equation (11) where normality is assumed for PC scores:

$$0 \pm t_{(I-1,\alpha/2)} s_r (1 + 1/I)^{1/2} \tag{11}$$

where $t$ is the student $t$-distribution with $I$-1 degrees of freedom at a significance level of $\alpha$, and $sr$ is the standard deviation of $r$th PC scores based on NOC data. The 0 is due to the mean-centering operation as part of the auto-scaling of data, making the expected value equal to zero.

### 2.3. Contribution plots

A contribution plot shows the importance or influence of a process variable in a statistic (SPE, D-statistics, or score value). Once a fault is exposed according to one or more control charts, the corresponding contribution plot can be examined to detect the process measurements that were affected by the fault with the hope of identifying a possible cause for the AOC behavior of a production batch; as a result proper action can be taken to bring the process back to the control region for the next production run. Contributions of variable $j$ to the SPE, D-statistic, and a PCA score at time point $k$ are respectively calculated as [3,5,20]:

$$C_{ijk}^{SPE} = e_{ijk}^2 \tag{12}$$

$$C_{ijk}^{D} = \sum_{r=1}^{R} S_{(r)(r)}^{-1} t_{i(r)} x_{ijk} p_{jk(r)} \tag{13}$$

$$C_{ijk}^{t_r} = x_{ijk} p_{jkr} \tag{14}$$

Contributions to SPE are always positive as they are squared residuals. Contributions to the D-statistic and score values can be either positive or negative, as seen from Equations (13) and (14). Traditionally CLs are not used for the contribution plots because there is no standard asymptotic procedure to estimate them. Therefore those variables with the largest positive contributions are habitually investigated for fault diagnosis for both SPE- and D-statistic [5]. However, Westerhuis et al. have suggested using an asymptotic recipe – the same formulas to estimate CLs for SPE charts presented in Equations (5)–(7) – to build CLs for SPE contributions. These authors also suggested leave-one-out jackknifing to build CLs for contributions to D-statistic [5]. The strategy by Westerhuis will be compared with our bootstrap-based CLs approach in *Results and discussion* section to evaluate which one performs better.

To build contribution plots for the D-chart two technical decisions need to be made. The first one addresses how to use the data to compute contributions. In case of a local model data of only the current time point should be used since this data selection is applied to compute the D-statistic. In case of an evolving model either data of only the current time point or all the data available up to the current time point can be used to compute the contributions. The user has to decide which one is more appropriate in relation to the process at hand. It might be reasonable to use the data of only the current time point when diagnosis of sudden changes is of interest (e.g. sudden malfunctioning of a sensor), while it might make more sense to use all the data available when the overall status of the process in identification of a fault is desired (e.g. a systematically deviating time-profile in a AOC batch run). The latter seems to be the more natural choice, which was applied in this study, because the D-statistic (based on an evolving model) is estimated using also all the data available up to the current time point, potentially making the calculated contributions conform better to the statistic. The second decision that has to be made in building CLs for D-chart contribution plots is how to deal with negative contributions. One option is to use the absolute values of contributions as was done by Conlin et al. [4]. In the approach by Westerhuis et al. both negative and positive contributions were used to build CLs for D-contribution plots, but only the upper CL was used for detection during process monitoring [5]. It was mentioned by the authors that the lower CL was not used because only high contributions force the D-statistic to be out of control. This works fine for offline [5], one-model based [1], and evolving model MSPC because the sign of scores is constant for the first two strategies and nearly constant for the last one over all the time points due to the evolving nature of the model, where the model for each time point is fitted to only slightly changing data (owing the nature of batch processes). A constant sign for scores leads to a constant sign for the contribution which makes only high positive contributions possible and allows us to ignore the lower CLs. In case of a strategy such as a local model, an independent model is built for each time point where the sign of the scores can change arbitrarily which causes changes in the sign of the contribution. This often leads to high negative contributions. It might therefore make more sense to use either absolute contributions or negative contributions in combination with lower CLs, especially when the sign of contributions is important in fault diagnostics. The last choice is the approach taken in our work for the local model while lower CLs are not used for the evolving model, based on the suggestion by Westerhuis et al.

It should be mentioned here that an indirect way of determination of faulty variables for D-statistic through inspection of contributions to PC scores has been suggested previously in literature [21]. In this strategy, once a fault is detected in the D-chart, normalized scores (squared scores values divided by the singular values for each PC) are inspected. Those scores with high normalized values are identified as suspicious. Contribution plots for each detected score are investigated and highly contributed variables with the same sign (compared to the corresponding scores) are determined as the faulty variables. This somewhat ad hoc method is not as straightforward as direct inspection of contributions to D-statistic and not pursued in our work.

In case a fault is detected in the score chart, the largest contributions that have the same sign as the scores detected in the score control chart should be investigated for fault diagnosis [2,3].

## 3. Bootstrap confidence limits in MSPC and contribution statistics

Non-parametric re-sampling is a common way to apply the bootstrap for uncertainty estimation [6,7,22]. In this procedure *B* bootstrap sample sets are drawn randomly, with replacement, from objects in the empirical data (in our case NOC batches). Bootstrap estimates of scores and loadings are obtained by applying PCA to the bootstrap sample sets. Reordering and sign reflection might be needed to correct bootstrap scores and loadings as an inversion of PCs and a combined sign flip for scores and loadings might occur because of re-sampling perturbations [22–24]. In addition to the two previously mentioned effects, the PCA solution from a bootstrap sample might be a rotated version of the PCA subspace of the empirical data [25]. It has been suggested to rotate each bootstrap estimated subspace towards the empirical estimate subspace by applying Orthogonal Procrustes rotation so as to not inflate the uncertainty to an unrealistic magnitude [25]. This step is debatable as some people see it as a contradiction to the concept of resampling and bootstrapping. In practice this step has been shown to not make much of a differences in the final estimated CLs [26]. Once the bootstrap scores and loadings are corrected, SPE, D and individual score statistics can be calculated.

The so-called Bias-Corrected and Accelerated ($BC_a$) method has been reported as the most accurate method to build CLs from bootstrap estimates in the literature [6,7,27] as it accounts for the bias and the skewness of the distributions in estimating the CLs; therefore it was used in this study. The procedure uses two auxiliary parameters: $\widehat{z}_0$ to correct for bias (difference between a parameter estimate $\widehat{\theta}$ - e.g. the SPE-, D-statistic or Contributions - and the mean of all bootstrap-based $\widehat{\theta}_{.}$ estimate values, $\widehat{\theta}^*_{.}$), and $\widehat{\alpha}$ to take skewness of the distribution into account (see Ref. [8] for implementation questions and [23] for further details). According to the $BC_a$ recipe lower and upper CLs for any parameter $\widehat{\theta}$ are found as $\widehat{\theta}^{*(\alpha_1)}$ and $\widehat{\theta}^{*(\alpha_2)}$ where:

$$\alpha_1 = \Phi\left(\widehat{z}_0 + \frac{\widehat{z}_0 + z^{\alpha}}{1 - \widehat{\alpha}(\widehat{z}_0 + z^{\alpha})}\right) \tag{15}$$

$$\alpha_2 = \Phi\left(\widehat{z}_0 + \frac{\widehat{z}_0 + z^{(1-\alpha)}}{1 - \widehat{\alpha}(\widehat{z}_0 + z^{(1-\alpha)})}\right) \tag{16}$$

with Φ being the standard normal cumulative distribution function, while $z^{\alpha}$ is the $100\alpha^{\text{th}}$ percentile point of a standard normal distribution. $\widehat{z}_0$ is calculated as:

$$\widehat{z}_0 = \Phi^{-1}\left(\frac{\#\left[\widehat{\theta}^*_b < \widehat{\theta}\right]}{B}\right) \tag{17}$$

where $\Phi^{-1}$ is the inverse of the standard normal cumulative distribution function, $\widehat{\theta}^*_b$ is the *b*th bootstrap parameter estimate, and $\widehat{\theta}$ is the empirical estimate. The numerator in the parenthesis translates as the number of $\widehat{\theta}^*_b$ values that are less than $\widehat{\theta}$ (out of *B* bootstrap re-samplings). One common way to estimate $\widehat{\alpha}$ is to perform jackknifing on the empirical data [6]:

$$\widehat{a} = \frac{\sum_{i=1}^{I}\left(\widehat{\theta}_i - \widehat{\theta}_{.}\right)^3}{6\left[\sum_{i=1}^{I}\left(\widehat{\theta}_i - \widehat{\theta}_{.}\right)^2\right]^{3/2}} \tag{18}$$

where $\widehat{\theta}_i$ is the estimated parameter from the empirical data with the *i*th row removed (so-called leave-one-out jackknifing), and $\widehat{\theta}_{.}$ is the mean value of all the jackknife estimates. Since the CLs in this work are based on predictions (see Ref. [9] for more details) leave-one-out cross-validation is applied, instead of leave-one-out jackknifing, to estimate $\widehat{\alpha}$. After estimation of $\alpha_1$ and $\alpha_2$ the lower and upper CLs are set as the $B\alpha_1^{\text{th}}$ and $B\alpha_2^{\text{th}}$ ordered value of *B* bootstrap estimates; note that $B\alpha_1$ and $B\alpha_2$ need to be rounded to the nearest integers.

The complete bootstrap recipe to build prediction-based CLs for the control charts, including contributions, is thus [9]:

1. Generate *B* non-parametric bootstrap sample sets $\mathbf{X}^{\mathbf{b}}(I \times JK)$ by random selection with replacement from NOC/empirical batches
2. Center and scale (auto-scale) the empirical data and bootstrap samples
3. Apply PCA to the empirical data and bootstrap samples
4. Reorder and reflect (and apply Procrustes rotation on) bootstrap scores and loadings using the empirical score and loadings as

targets (Procrustes rotation is optional here as it has only a marginal contribution to estimated CLs)

5. Objects that are not present in each bootstrap sample are projected onto (hence predicted by) the model for that bootstrap sample (in a cross-validation like fashion)
6. Calculate SPE and D-statistic for all predicted objects
7. Build $BC_a$ CLs from bootstrap-predicted estimates for each statistic and the corresponding process variable contributions using Equations (15)–(18). $\widehat{\theta}$ for a statistic, used in Equation (17), can be set to the mean or median of the cross-validated estimates of the statistic.

## 4. Data

The well-known batch dataset of polymerization of Nylon 6′6 from DuPont Co. was used in this study. It includes data from 55 batches ($I$), where 10 process variables ($J$) were measured in 100 time points ($K$) for each batch (see Refs. [1,15] for details). Batches 1 through 36 were considered as NOC batches according to the study by Nomikos et al. [1] and were used to build control charts and contribution plots with confidence limits. The remaining 19 batches were considered as AOC.

## 5. Results and discussion

The two MSPC recipes were applied to the data of 36 NOC batches where the number of PCs for each method was determined by cross-validation: 2 for the local model, and 3 for the evolving model. It is important to note that the number of PCs should be selected based on the performance of the model in detection of abnormal variations (as the main goal in MSPC) that is the lowest overall type II error while overall type I error is predefined. In case evaluation of such performance is not feasible cross-validation seems a reasonable choice. To calculate bootstrap CLs, 2000 non-parametric bootstrap sample sets were generated. This decision was based on Davison and Hinkley's rule of thumb $B \approx 40\,I$ ($I$ is the empirical sample size) bootstrap samples in estimation of confidence interval. Sufficiently large $B$ decreases the Monte Carlo error of the re-sampling (the difference in the outcome when the complete procedure of uncertainty estimation is repeated), which therefore increases the accuracy of the bootstrap approximation [7]. SPE and D-statistic were calculated for empirical and bootstrap samples, and 95% and 99% $BC_a$ CLs were built from the bootstrap-predicted estimates of the statistics and the corresponding contribution plots.

Fig. 1 shows the distribution of one process variable (number 8 – a pressure) contribution to SPE, D-statistic, and scores estimated by bootstrapping on data of the NOC batches based on the evolving model.

As can be seen, the contributions to SPE are asymmetric and positive, while contributions to the D-statistic have resulted in an asymmetric distribution where a small fraction of contributions are negative (an expected artifact of the method of computation [5]). Contributions to the PC1 and PC2 score do not follow a specific distribution, but rather an ensemble of what appear to be four distributions, while the distribution for the PC3 score seem to follow an approximately normal distribution. Process data usually come from designed and controlled spaces that are naturally affected by factors with non-random characteristics. Since the principal components in PCA represent these process factors it is reasonable (and often observed) to see the contributions to the first PC not following a normal distribution. The noise along PCs is structured as it is minimized for the first PC and becomes relatively larger for successive PCs because of the embedded nature of PCA. This makes the contributions to the first PC unlikely to follow a normal distribution, while higher PCs follow the normal distribution closer where the signal-to-noise ratios associated with them become smaller.

An alarm is raised when one or more statistics pass the 95% limit while a fault in a process is reported when a batch exceeds the 99% limit in a MSPC control chart. Once an alarm or fault is detected, it is important to find the process variable(s) that was (were) affected by it. This assists the process operators and optimization engineers in their cause-and-effect analysis needed to bring the process back into the region of *statistical control*. This optimization towards NOC procedure is obviously facilitated by inspecting only the affected variables or process part instead of all possible process tags, and contribution plots can help to find the variables affected by an alarm or fault. However, having a high contribution does not automatically mean a variable marks a fault since process variables in general contribute unequally to the statistics just as they contribute unequally to the model and residuals, even when scaled to unity (equal initial) variance. This is because each variable is controlled and treated differently during the process, and the signal-to-noise ratio associated with each variable is potentially different from the others. Thus, simply assigning the largest contributing variables as possible faulty variables might result in mistakes purely due to the scale differences in natural variation, measurement uncertainty, etc. Comparison of contributions for new batches with contributions for NOC batches [4,5] is a better way to find true faulty variables, where CLs for contribution plots (estimated from NOC batches) could be a reasonable criterion to judge new batches.

Batch #49, which resulted in borderline product quality, will be used here as AOC batch to investigate the performance of contribution plots with bootstrap CLs for the two MSPC recipes. SPE charts signaled a clear fault at time point 57 for both of the recipes (Fig. 2).

The detected faulty period was from time point 57 through 64 for the local model, while it was present from 57 through 65 for the evolving model. The corresponding contribution plots at time point 57 were examined to identify the faulty process variables using the bootstrap CLs. Fig. 3 shows the results for both recipes, applying once CLs based on asymptotic estimation (Fig. 3a,c) [5] and once with CLs based on bootstrapping (Fig. 3b,d).

Comparison of the two types of CLs for the local model (Fig. 3a,b) show that the bootstrap CLs are slightly higher. Nevertheless, both types of CLs suggest that all the process signals except variables 1, 3 and 10 can be assigned as faulty variables when compared to NOC results for this particular stage in the batch. Contribution plots for the evolving model (Fig. 3c,d) show that the two types of CLs are similar and both suggest variables 6 to 10 as faulty. Local and evolving recipes do not suggest the same set of variables as the faulty because they use a different model structure (in the evolving model the history of the batch is taken into account, while the local model does not). The data of batch #49 can be compared visually with the data of NOC batches to see which recipe gives more serviceable results (Fig. 4).

The figure shows that the deviation from NOC in batch #49 can be observed most clearly in the time-trends of variables 6 to 10 where the values decrease unexpectedly at time point 57, while this is not seen in the data of NOC batches. An evolving model takes the evolution of batches into account by modeling the batch data up to the current time point, where the SPE chart implicitly accounts for changes in the trajectory of the data even though the computation uses the residuals associated with only the current time point. This capability of the evolving model recipe might be the reason of more interpretable results for this particular example.

The simultaneous deviations in four variables (6, 7, 8, and 9)

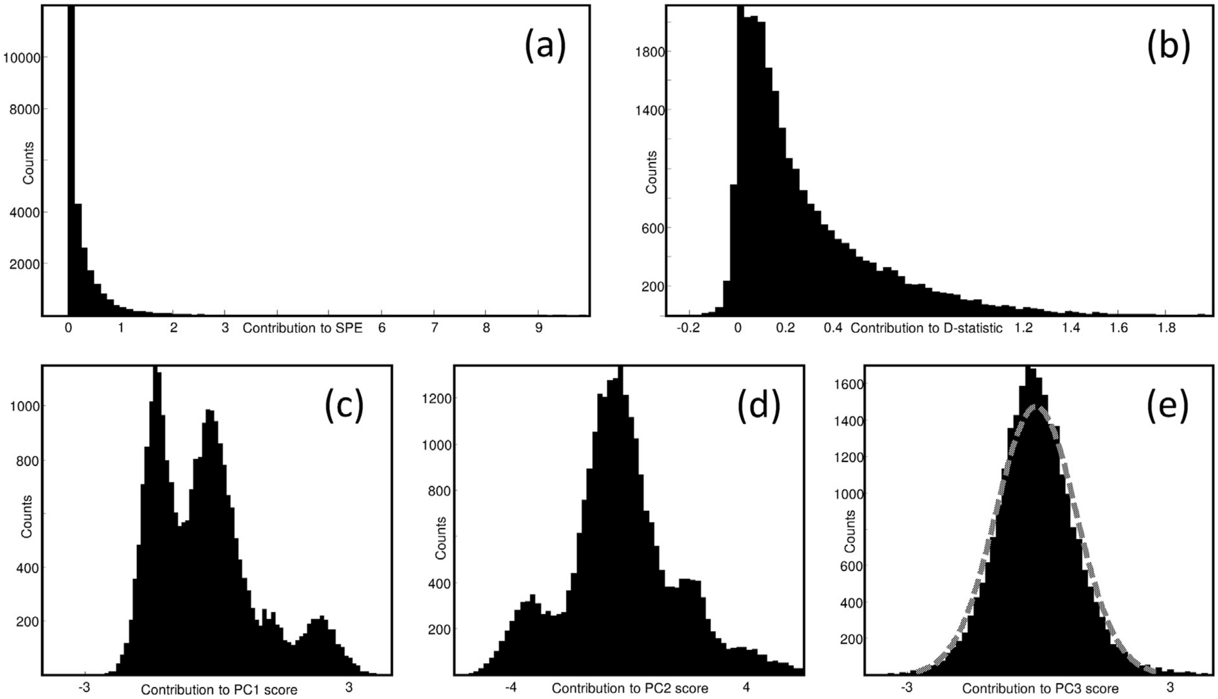


**Fig. 1.** Distributions of contributions of process variable 8 (a pressure) at the last time point for (a) SPE; (b) D-statistic; (c) PC1 score; (d) PC2 score; (e) PC3 score (with a normal distribution superimposed) based on the evolving model MSPC strategy estimated by $BC_a$ bootstrapping.

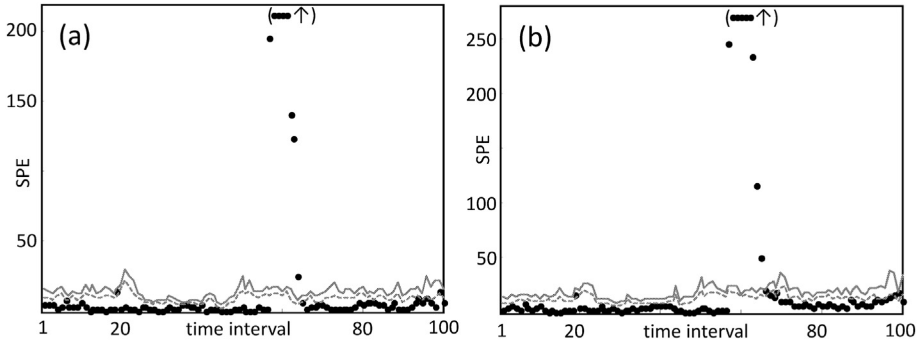


**Fig. 2.** Monitoring of batch #49 using online SPE charts with bootstrap CLs ("–" 95%, "—" 99%) (a) based on the local model; (b) based on the evolving model.

from NOC average trajectories were also concluded to be the cause of the detected fault for batch #49 by Nomikos et al. using contribution plots [1]. It was mentioned in this reference that these four variables deviated in a systematic manner and then returned to NOC average trajectories, as seen in Fig. 4. Therefore, the special event in this batch was attributed to an operational problem with these process tags. It is interesting to note that only variables 6 to 9 were assigned as faulty, while in this work we also flagged the much smaller contribution of variable 10. Nomikos et al. [20] compared the data for batch #49 with the average batch trajectory, where the criterion for assigning faulty variables was the magnitude of deviation from the average trajectory. This is similar to looking at the magnitude of the contributions that gives variables 6 to 9 as the obvious answer to the problem since their deviations were very large compared to the other variables (see Fig. 3d). On the other hand, we used contribution plots with two types of CLs, where the criterion for assigning faulty variables is the magnitude of the contributions compared to NOC behavior under statistical control, including natural/accepted variance. This resulted in assigning variable 10 as faulty which matches with the data of this signal, where a deviation from NOC trajectories is clearly observed during the same time period as the other four faulty variables. This example illustrates that CLs in contributions can assist in diagnosing the faults. Further comparison of two types of CLs for contributions to SPE showed that they perform equally well with acceptable diagnostic power.

The D-statistic monitoring chart was also used to follow batch #49, illustrated in Fig. 5.

The D-chart based on the local model has detected the same faulty period as the SPE chart, while D-chart based on the evolving model has not recorded any faults - only a warning at time point 61. The D-chart based on an evolving model accounts for the whole batch history up to the current time point, where an abnormality is detected when the whole process deviates in a statistically significant way from the NOC trajectories. This makes the D-chart slow in detecting faults and insensitive to sudden perturbations, whereas the D-chart based on a local model responds fast since only the data of the current time point is used. Both positive and negative values of D-contribution were used to build contribution plots for both the local (Fig. 6a,b) and the evolving model (Fig. 6c,d) at time point 61

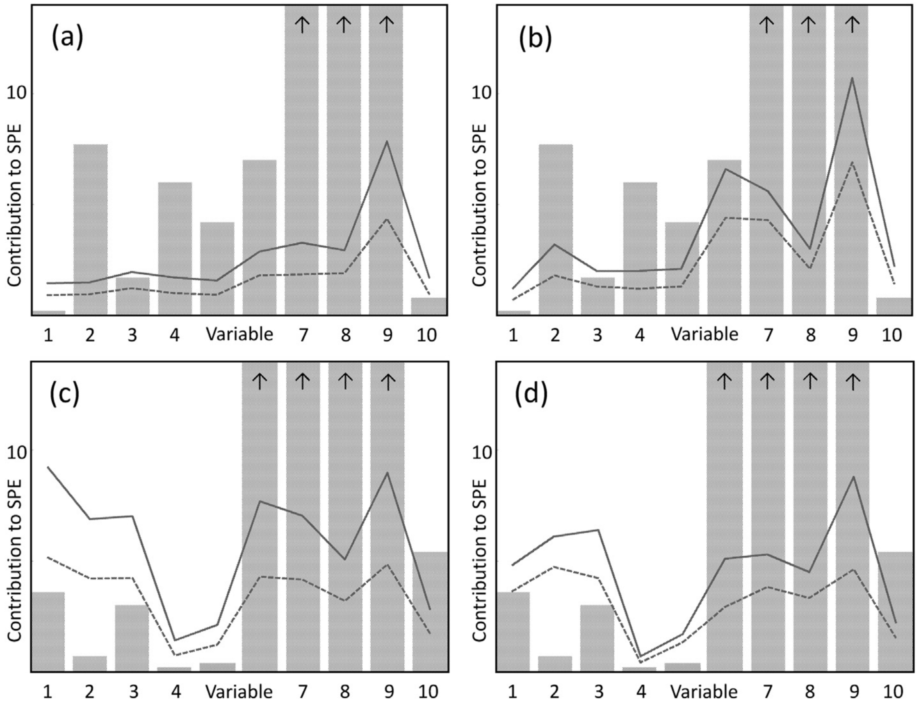


**Fig. 3.** Contributions to SPE for batch #49 at time point 57 (a) based on the local model with asymptotic CLs; (b) based on the local model with bootstrap CLs; (c) based on the evolving model with asymptotic CLs; (d) based on the evolving model with bootstrap CLs ("–" 95%, "—" 99%).

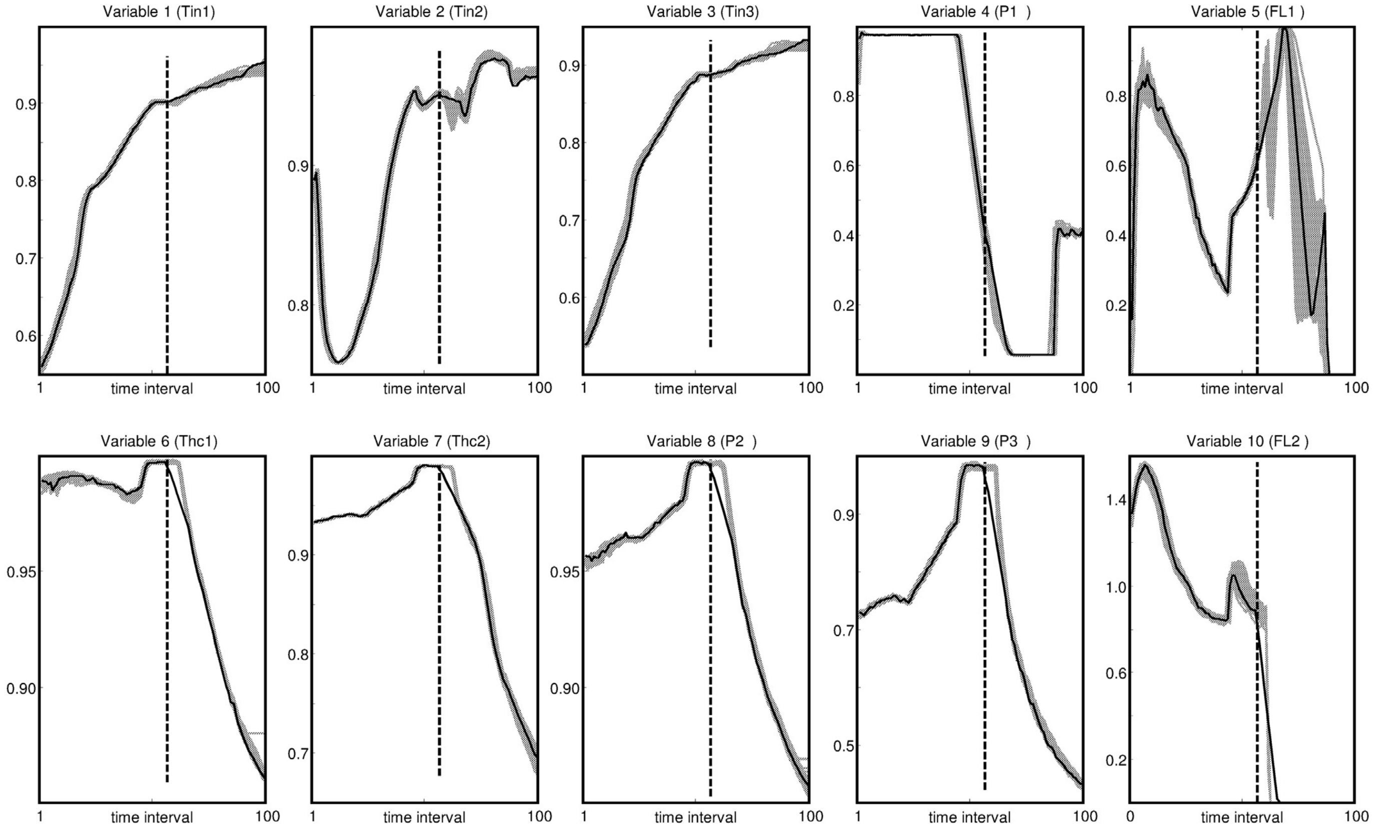


**Fig. 4.** Original data of the ten process variables for AOC batch #49 (solid line), and 36 NOC batches (gray "envelope"); the dotted line indicates time point 57.

as the common time point where an alarm was raised.

Contribution for the local model (Fig. 6a,b) illustrates that based on both types of CLs all the process variable were affected by the fault. Inspection of data at time point 61 shows that variable 1 to 5

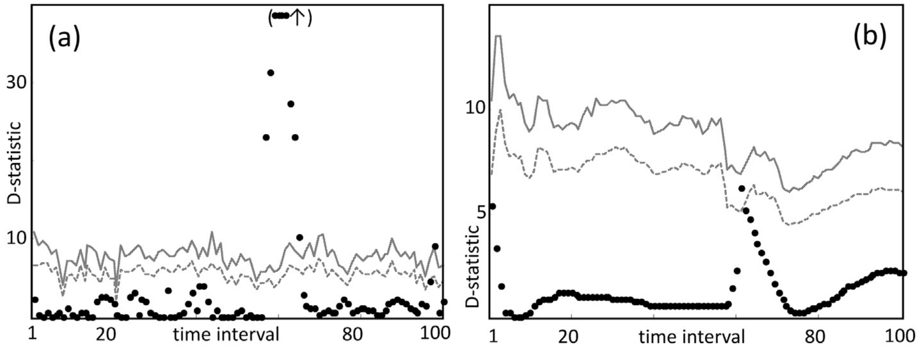


**Fig. 5.** Monitoring of batch #49 using online D-charts with bootstrap CLs ("–" 95%, "—" 99%) (a) based on the local model; (b) based on the evolving model.

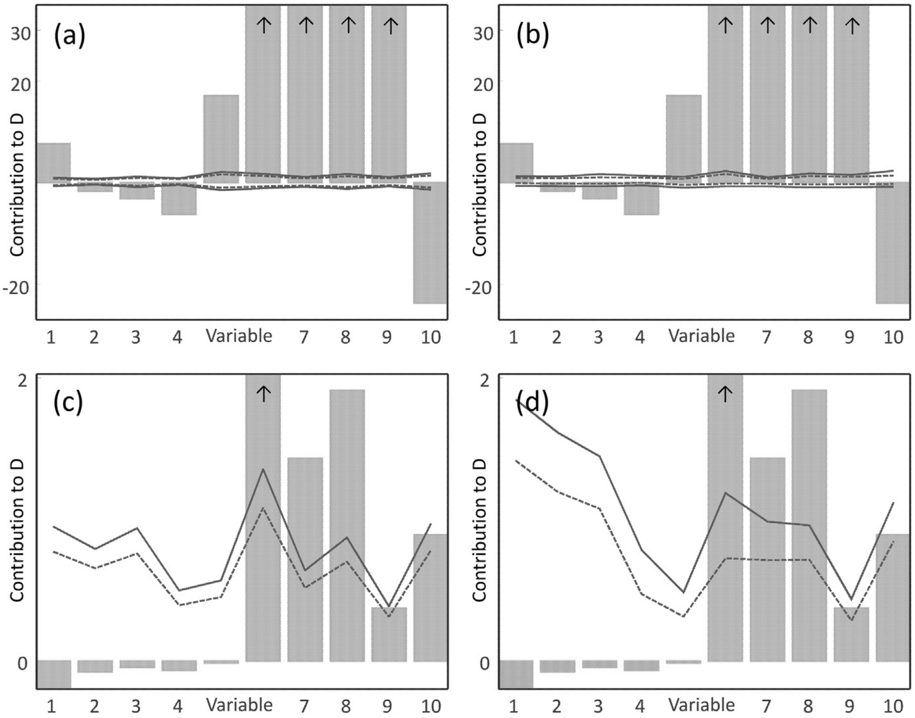


**Fig. 6.** Contributions to D-statistic for batch #49 at time point 57 (a) based on the local model with jackknife CLs; (b) based on the local model with bootstrap CLs; (c) based on the evolving model with jackknife CLs; (d) based on the evolving model with bootstrap CLs ("–" 95%, "—" 99%).

have small deviations from the NOC trajectory, while variable 6 to 10 have very large deviations. Since the local model is very sensitive to the NOC trajectory even small deviations caused by variable 1 to 5 were detected in the contribution plot. On the other hand, contribution plot for the evolving model (Fig. 6c,d) with both types of CLs shows only variable 6 to 10 as questionable responses using the CLs. Variable 1 to 5 were not diagnosed as faulty variables since again D-chart based on the evolving model is not very sensitive to small deviations. Comparison of CLs for the evolving model shows that for most of the variables the bootstrap CLs are much higher than jackknife CLs, which might mean that they are systematically different; more inspection is needed to confirm this matter. Note that for online MSPC schemes there can be *real-time* contribution plots for each variable for a specific statistic. These contribution plot can be used to make a better comparison of the two types of CLs. As an example of a faulty variable, *real-time* contribution of process variable 7 for the D-statistic based on the evolving model is illustrated for AOC batch #49 in Fig. 7, where contributions for all the time points are shown with jackknife and bootstrap CLs.

We can clearly see that the bootstrap CLs (Fig. 7b) are higher than jackknife CLs especially over the time range 60–100. The evolution of the contribution can be seen over time, where a faulty behavior is seen from time point 57 to around 60 which was also seen in the D-chart. The contribution starts to increase again at time point 71 were it exceeds both 95 and 99% jackknife CLs at time point 85 and stayed above the limits until the end of the batch run while it exceeded only 95% bootstrap CLs. Data inspection of this variable shows that the shape of the profile for this batch started to differ from the NOC shape at two times over the process: once at time point 71 (faster decreasing temperature) and another time at time point 80 (slower decreasing temperature); but the value remained well within the NOC range over the time range 71–100. The deviations caused an increase in the contribution, but they were not large enough to cause a fault and this has been handled better by the bootstrap CLs compared to the jackknife CLs, indicating that bootstrap CLs perform better for this observation.

*Real-time* contributions of process variable 9 (as another faulty variable) are shown in Fig. 8 to illustrate the difference between the

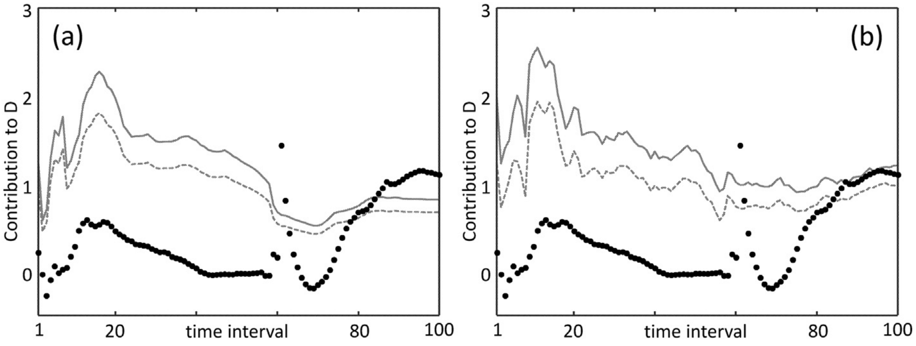


**Fig. 7.** *Real-time* contribution plots of process variable 7 (a temperature) to D-statistics for AOC batch #49 (.) based on the evolving model with (a) jackknife CLs; (b) bootstrap CLs ("–" 95%, "—" 99%).

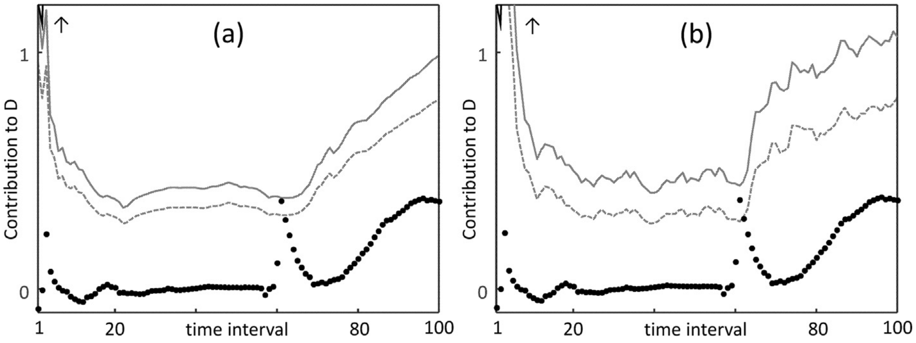


**Fig. 8.** *Real-time* contribution plots of process variable 9 (a pressure) to D-statistics for AOC batch #49 (.) based on the evolving model with (a) jackknife CLs; (b) bootstrap CLs ("–" 95%, "—" 99%).

two types of CLs.

Wider bootstrap CLs are seen also for variable 9 compared to jackknife CLs especially over the time range 60–90. Further comparison of the two types of CLs for other process variables (not shown) indicated that the differences are substantial for almost all process tags. In general bootstrap gave more reasonable CLs in comparison with the suggested recipe based on jackknifing [5], showing the capability of bootstrapping in CL estimation. On the other hand, leave-one-out jackknifing underestimates the uncertainty which has been already reported in other works [28].

Some general considerations need to be taken into account when contribution plots are used. The first point is to use the CLs in contribution plots only to find the potential variables that were affected by a fault, meaning that the CLs are for diagnosis purposes rather than statistical significance. The second argument is to focus on the contribution plots only when a warning has been given or a fault has been detected in the associated control chart. Using contribution plots (in an almost univariate way) when a fault has not been flagged might lead to misjudgment of the process performance since it is reasonable to see a contribution of one process variable exceeding the limits while the combination of all the variables (the multivariate monitoring) is still acceptable. We presented *real-time* contribution profiles in this work merely to compare the CLs since it was the most obvious way to compare their diagnostic performance, not as a suggested tool for daily practice. The third consideration is to keep in mind that a diagnosis cannot purely be based on a contribution plot; process knowledge and experience should always be employed in the diagnosis process, and lead to meaningful ways of tuning the process back into *statistical control*. However, the contribution profiles, accompanied by CLs, can be very valuable tools in performance evaluation and process optimization.

## 6. Conclusion

Bootstrap-based Confidence Limits for process variable contribution plots in online MSPC were presented. The idea is based on the previous thought to construct reliable bootstrap-based CLs for control charts [9]. The bootstrap CLs for contribution plots were compared with the previously suggested recipes based on asymptotic approximate for SPE and jackknifing for D-statistic where the bootstrap CLs showed two main advantages: 1) they give more reasonable results especially for contributions to D-statistic based on comparison of faulty and NOC contributions, where jackknifing has been shown to underestimate the uncertainty, 2) they are, unlike asymptotic CLs, available for all statistics and parameters, including variable contributions. These advantages are even more pertinent when a limited data set is available (e.g. at the start-up of a new process), and/or when the data set contains natural heterogeneity/clustering (e.g. when the same process is run in different unit operations or production locations).

## References

[1] P. Nomikos, J. MacGregor, Multivariate SPC charts for monitoring batch processes, Technometrics 37 (1995) 41–59.
[2] J. MacGregor, C. Jaeckle, C. Kiparissides, M. Koutoudi, Process monitoring and diagnosis by multiblock Pls methods, AIChE J. 40 (1994) 826–838.
[3] T. Kourti, J. MacGregor, Multivariate SPC methods for process and product monitoring, J. Qual. Technol. 28 (1996) 409–428.
[4] A. Conlin, E. Martin, A. Morris, Confidence limits for contribution plots, J. Chemom. 14 (2000) 725–736.
[5] J. Westerhuis, S. Gurden, A. Smilde, Generalized contribution plots in multivariate statistical process monitoring, Chemom. Intellig. Lab. Syst. 51 (2000) 95–114.
[6] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.
[7] A.C. Davison, D.V. Hinkley, Bootstrap Methods and their Applications,

Cambridge Univ. Press, Cambridge, 1997.

[8] R. Wehrens, H. Putter, L. Buydens, The bootstrap: a tutorial, Chemom. Intellig. Lab. Syst. 54 (2000) 35–52.

[9] H. Babamoradi, F. van den Berg, A. Rinnan, Comparison of bootstrap and asymptotic confidence limits for control charts in batch MSPC strategies, Chemom. Intellig. Lab. Syst. 127 (2013) 102–111.

[10] E. Martin, A. Morris, M. Papazoglou, C. Kiparissides, Batch process monitoring for consistent production, Comput. Chem. Eng. 20 (1996) S599–S604.

[11] E. Martin, A. Morris, An overview of multivariate statistical process control in continuous and batch process performance monitoring, Trans. Inst. Meas. Control 18 (1996) 51–60.

[12] A. Polansky, A general framework for constructing control charts, Qual. Reliab. Eng. Int. 21 (2005) 633–653.

[13] F. Wang, Y. Eldon, Confidence intervals in repeatability and reproducibility using the Bootstrap method, Total Qual. Manag. Bus. Excell 14 (2003) 341–354.

[14] P. Phaladiganon, S.B. Kim, V.C.P. Chen, J. Baek, S. Park, Bootstrap-based T2 multivariate control charts, Commun. Stat. Simul. Comput. 40 (2011) 645–662.

[15] K. Kosanovich, K. Dahl, M. Piovoso, Improved process understanding using multiway principal component analysis, Ind. Eng. Chem. Res. 35 (1996) 138–146.

[16] H. Ramaker, E. van Sprang, J. Westerhuis, A. Smilde, Fault detection properties of global, local and time evolving models for batch process monitoring, J. Process Control 15 (2005) 799–805.

[17] S. Qin, Statistical process monitoring: basics and beyond, J. Chemom. 17 (2003) 480–502.

[18] J. Jackson, G. Mudholkar, Control procedures for residuals associated with principal component analysis, Technometrics 21 (1979) 341–349.

[19] N. Tracy, J. Young, R. Mason, Multivariate control charts for individual observations, J. Qual. Technol. 24 (1992) 88–95.

[20] P. Nomikos, Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis - World Batch Forum, Toronto, May 1996, ISA Trans. 35 (1996) 259–266.

[21] T. Kourti, Application of latent variable methods to process control and multivariate statistical process control in industry, Int. J. Adapt. Control Signal Process 19 (2005) 213–246.

[22] H. Babamoradi, F. van den Berg, Å. Rinnan, Bootstrap based confidence limits in principal component analysis — a case study, Chemom. Intellig. Lab. Syst. 120 (2013) 97–105.

[23] P. Peres-Neto, D. Jackson, K. Somers, Giving meaningful interpretation to ordination axes: assessing loading significance in principal component analysis, Ecology 84 (2003) 2347–2363.

[24] M.E. Timmerman, H.A.L. Kiers, A.K. Smilde, Estimating confidence intervals for principal component loadings: a comparison between the bootstrap and asymptotic results, Br. J. Math. Stat. Psychol. 60 (2007) 295–314.

[25] L. Milan, J. Whittaker, Application of the parametric Bootstrap to models that incorporate a singular-value decomposition, Appl. Stat. J. R. Stat. Soc. 44 (1995) 31–49.

[26] H. Babamoradi, Bootstrap-based Confidence Estimation in PCA and Multivariate Statistical Process Control, Doctoral Dissertation, University of Copenhagen, Denmark, 2012.

[27] B. Efron, Better bootstrap confidence-intervals, J. Am. Stat. Assoc. 82 (1987) 171–185.

[28] A. Rinnan, D. Giacalone, M.B. Frost, Check-all-that-apply data analysed by partial least squares regression, Food. Qual. Prefer. 42 (2015) 146–153.