Computers and Chemical Engineering 208 (2026) 109585

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Partial least-squares model adaptation by bootstrap resampling

Elia Arnese-Feffin [a,1], Jinwook Rhyu [a,1], Benjamin T. Smith [b], Chris D. Castro [b], Jacqueline M. Wolfrum [c], Stacy L. Springs [c], Roger A. Hart [b], Tom Mistretta [b], Richard D. Braatz [a,c,*]

[a] Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, United States of America
[b] Amgen, One Amgen Center Drive, 91320 Thousand Oaks CA, United States of America
[c] Center for Biomedical Innovation, Massachusetts Institute of Technology, 77 Massachusetts Avenue, 02139 Cambridge, MA, United States of America

ARTICLE INFO

Dataset link: https://github.com/EliaAF/PLSBootstrapAdaptation

Keywords:
Partial least-squares regression
Model adaptation
Soft sensor
Product quality prediction
Biomanufacturing

ABSTRACT

Soft sensors play a crucial role in biomanufacturing, the prime example being the estimation of product quality attributes using only easy-to-measure variables from the plant instrumentation. Data-driven models, such as partial least-squares regression, are widely used to this end. However, their predictive performance might degrade whenever process conditions shift, even after changes made on purpose by operators. Model adaptation strategies can keep the model up to date, but they tend to under-deliver when the process undergoes abrupt changes. In this study, we propose a novel model adaptation method that exploits knowledge of intentional abrupt changes in the process via bootstrap resampling. Exploiting such knowledge, the available data can be used in an optimal way. We demonstrate our approach on three case studies: a numerical example, a simulated penicillin production process, and an industrial biomanufacturing process. Compared to existing adaptation strategies, our method achieves faster adaptation and better predictive performance in the transition period between old and new process conditions. The proposed approach thus enables practitioners to quickly recover the predictive power of soft sensors, with clear benefits on process operation and product quality.

1. Introduction

Ensuring product quality in biomanufacturing processes is fundamental to guarantee the safety and effectiveness of biopharmaceutical products. This is achieved by constant monitoring of the CCPs and CQAs of the production process, i.e., product quality attributes and process variables, respectively. CCPs are typically measured online at a high sampling rate, thereby guaranteeing appropriate resolution for process monitoring and prompt fault detection. However, tens or hundreds of process variables are typically recorded, challenging traditional univariate monitoring methods (Reis and Gins, 2017). On the other hand, CQAs are fewer in number, but often involve time-consuming and expensive measurement procedures, thus fewer observations are available (O'Flaherty et al., 2020). This makes online quality monitoring challenging.

Data-driven and machine learning models can help tackle the aforementioned challenges. These models exploit data collected in production processes and have proved to be valuable assets in online CQA prediction (Hong et al., 2023; Mohr et al., 2024) and process monitoring (Ündey et al., 2004). Latent variable methods, such as PCA (Wold et al., 1987; Wise and Gallagher, 1996), exploit correlation among process variables to reduce the dimensionality of the data, yet still retain a comprehensive, multivariate monitoring approach by providing suitable monitoring statistics (Chiang et al., 2001). Soft sensors (Kadlec et al., 2009; Zhu et al., 2020) can be used to aid quality monitoring by providing real-time predictions of hard-to-measure CQAs based on easy-to-measure process variables. In the context of biomanufacturing, PLS regression (Geladi and Kowalski, 1986; Wold et al., 2001) is widely used, enabling both prediction of the CQAs and multivariate process monitoring.

PLS models can be developed based on historical production data. However, the model captures only a "snapshot" of the process, i.e., the information in the dataset at hand. Prediction and monitoring performance could degrade upon process changes, e.g., slow membrane fouling in downstream processing or variations of the culture medium composition in upstream processing (Lima et al., 2022). Changes in the process scale (e.g., a variation in the culture volume in a batch bioreactor) can be particularly detrimental to the model performance due to complex changes in the mixing patterns, gas–liquid mass transfer, and,

* Correspondence to: 77 Massachusetts Avenue, Room E19-551, Cambridge, MA 02139, United States of America.
E-mail address: braatz@mit.edu (R.D. Braatz).
[1] These authors contributed equally.

https://doi.org/10.1016/j.compchemeng.2026.109585
Received 25 September 2025; Received in revised form 29 January 2026; Accepted 2 February 2026
Available online 2 February 2026

ultimately, biological behavior. These are well-known challenges in the scale-up of biomanufacturing processes (Facco et al., 2020).

Model adaptation methods (Kadlec et al., 2011) provide a way to handle the changing nature of a process. Traditional strategies, such as moving-window and exponential-forgetting, can be combined with PLS modeling to design recursive algorithms for model update (Dayal and MacGregor, 1997; Qin, 1998). Such methods are easy to implement and agnostic to the kind of change affecting the process, thus being attractive solutions for model adaptation. However, their performances vary for different modes of change, e.g., slow drifts or abrupt steps. In the case of a step change in the data, traditional model adaptation methods do not typically perform well immediately after the change due to the strong imbalance between "old" and "new" data conditions (Chu et al., 2021). In fact, the new data could act as outliers in the dataset, which is still dominated by the old data. A similar issue applies when a few old observations are left in the memory of the model, thus adaptation is typically completed only after all old observations have been forgotten. Finally, care must be taken to avoid forgetting valuable information stored in past data when the process operates steadily, i.e., no change is happening.

A key point of traditional adaptation methods is that they typically do not make any assumption on the nature of the process change or on its location in time, i.e., they are *blind* adaptation strategies (Gama et al., 2014). In fact, such methods can be executed even if no process change at all is involved. On the other hand, some process changes are known *a priori* or can be easily detected, e.g., the culture volume of a bioreactor can be increased or decreased in response to a change in the product demand. Such valuable information, i.e., the presence of a well-defined change in the process, should be exploited in *informed* adaptation strategies. This strategy is at the heart of the so-called transfer learning paradigm (Chu et al., 2021; Briceno-Mena et al., 2023).

In this study, we propose an informed model adaptation strategy tailored to tackle known abrupt changes in the process. Our method combines PLS modeling with bootstrap resampling (Efron, 1979; Efron and Tibshirani, 1993) to accelerate the model adaptation. Specifically, we use the bootstrap to artificially augment the dataset comprising new conditions after the change, thus avoiding the aforementioned data imbalance issue. We do so by sampling with replacement from the new data to obtain a reasonable number of observations, then corrupt each observation by adding Gaussian noise with zero mean and covariance consistent with the noise in the old data (estimated using a PLS model of the old data alone) to obtain a realistic augmented dataset. Additionally, we perform several repetitions of bootstrap resampling to obtain multiple augmented datasets, calibrate multiple PLS models, and, consequently, obtain multiple estimates of the CQA. In this way, we obtain both robust estimates of the CQAs and a quantification of the prediction uncertainty.

The remainder of this article is organized as follows. We introduce the mathematical methods relevant to this work in Section 2, and we outline the proposed approach in Section 3. In Section 4, we demonstrate the proposed approach on three case studies: a simple numerical case study, a simulated bioreactor case study based on the IndPenSim model (Goldrick et al., 2015), and an industrial biomanufacturing process. Finally, we draw the conclusions of this study in Section 5.

2. Mathematical methods

In this Section, we introduce the mathematical methods used in this study, namely PLS regression and model adaptation strategies. The proposed model adaptation method is introduced in Section 3.

2.1. Partial least-squares regression

PLS (Geladi and Kowalski, 1986; Wold et al., 2001) provides a regression model between a matrix of input variables $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$, gathering N observations of V_X CCPs, and a matrix of output variables $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^{V_Y}$, collecting the corresponding observations of V_Y CQAs. Columns of matrices $\mathbf{X}$ and $\mathbf{Y}$ are assumed to be mean-centered and possibly scaled to unit variance. PLS also performs dimensionality reduction of both the input and output spaces, as defined by the matrix decomposition models

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \tilde{\mathbf{X}}, \tag{1}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^\top + \tilde{\mathbf{Y}}, \tag{2}$$

where $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are the reconstruction residuals of $\mathbf{X}$ and $\mathbf{Y}$, respectively, $\mathbf{P} \in \mathbb{R}^{V_X} \times \mathbb{R}^A$ and $\mathbf{Q} \in \mathbb{R}^{V_Y} \times \mathbb{R}^A$ are the input and output loading matrices, respectively, and $\mathbf{T} \in \mathbb{R}^N \times \mathbb{R}^A$ and $\mathbf{U} \in \mathbb{R}^N \times \mathbb{R}^A$ are the corresponding score matrices. A is the number of latent variables in the PLS model, i.e., the dimensionality of the reduced spaces, which is typically much lower than V_X.

Input and output latent variables are formulated to maximize the cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$ captured by the model, which is achieved by establishing a linear regression model in the latent space, with $\mathbf{T}$ as predictor matrix and $\mathbf{U}$ as response matrix. Columns of the $\mathbf{T}$ matrix are built to maximize their covariance with corresponding columns of the $\mathbf{U}$ matrix, while at the same time retaining as much variance of $\mathbf{X}$ as possible. This causes the columns of $\mathbf{T}$ to lie in a space of latent variables common to both $\mathbf{X}$ and $\mathbf{Y}$. When applying PLS as a regression model, the input variables are projected onto the space of latent variables,

$$\mathbf{T} = \mathbf{X}\mathbf{R}, \tag{3}$$

where $\mathbf{R} \in \mathbb{R}^{V_X} \times \mathbb{R}^A$ is the rotation matrix of $\mathbf{X}$, defined as $\mathbf{R} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}$, and $\mathbf{W} \in \mathbb{R}^{V_X} \times \mathbb{R}^A$ is the weight matrix of $\mathbf{X}$. Output variables can then be predicted by reconstruction from the projected predictor variables by using the reconstruction model

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^\top + \mathbf{H}, \tag{4}$$

where $\mathbf{H}$ is the matrix of prediction residuals of $\mathbf{Y}$. The columns of $\mathbf{W}$ and $\mathbf{Q}$ are computed pairwise by maximizing the cross-covariance between the input variables and the output variables explained by each one of the latent variables. For instance, the first pair is obtained by solving the optimization

$$\begin{aligned} &\arg\max_{\mathbf{w}_1,\mathbf{q}_1} \mathbf{w}_1^\top \mathbf{X}^\top \mathbf{Y} \mathbf{q}_1 \\ &\text{s.t. } \mathbf{w}_1^\top \mathbf{w}_1 = \mathbf{q}_1^\top \mathbf{q}_1 = 1. \end{aligned} \tag{5}$$

The columns of matrices $\mathbf{T}$, $\mathbf{U}$, $\mathbf{R}$, and $\mathbf{P}$ are constructed after the solution of each optimization. Detailed descriptions of PLS are given by Geladi and Kowalski (1986) and Wold et al. (2001). The number of latent variables, A, is typically set to maximize the prediction performance of the model. Cross-validation is a popular method to optimize A (Bro et al., 2008; Louwerse et al., 1999).

A calibrated PLS model can also be used for process monitoring, e.g., detecting faults or changes in the CCPs affecting the CQAs, which might signal the need for model adaptation. Two statistics can be used to this end: the T^2 statistic measures the squared distance of the current process state from the average conditions; the Q statistic measures the squared distance of the current process state from the space of latent variables. We refer the reader to the literature for mathematical details on the statistics and on their significance tests (Chiang et al., 2001; Qin, 2003; Mohr et al., 2025).

2.2. Model adaptation methods

Model adaptation strategies are meant to update data-driven models in changing process conditions scenarios, which are reflected in process data, e.g., as drifts or abrupt step changes in time. Traditional adaptation strategies are based on instance selection or instance weighting, which translate into moving-window and exponential-forgetting adaptation, respectively (Kadlec et al., 2011). These strategies are general and not dependent on the model, in that they operate at the data level.

In moving-window adaptation, the data-driven model at time t is built using the most recent K observations. The training dataset at time t is thus $\mathcal{D}_t = \{(\mathbf{x},\mathbf{y})_{t-K+1}, \dots, (\mathbf{x},\mathbf{y})_t\}$, where $(\mathbf{x},\mathbf{y})_t$ is the couple of input and output vectors, respectively, observed at time t. The data-driven model is typically adapted (i.e., re-trained) on dataset $\mathcal{D}_t$ after a new observation has been acquired. Alternatively, the adaptation is performed block-wise, i.e., after a given number of new observations has been acquired.

Exponential-forgetting adaptation operates by recursively discounting information in old observations, thus weighting more heavily recent observations in model training. The training dataset at time t includes all past observations: $\mathcal{D}_t = \{(\mathbf{x},\mathbf{y},w)_0, \dots, (\mathbf{x},\mathbf{y},w)_t\}$. The data include an additional scalar, $w_t \in [0,1]$, which is the weight of observation $(\mathbf{x},\mathbf{y})_t$ in model training. At time t, the weight of observation $(\mathbf{x},\mathbf{y})_{t-k}$, with $k \in \{0,\dots,t\}$, is computed as $w_{t-k} = \lambda^k$, where $\lambda \in [0,1]$ is the forgetting factor. Therefore, $w_t = 1$, $w_{t-1} = \lambda$, $w_{t-2} = \lambda^2$, and so on. The weight of observations follows an exponential decay (as $\lambda < 1$), hence the name "exponential-forgetting adaptation". Note that this strategy requires the use of training algorithms compatible with observation weighting, e.g., weighted least-squares instead of ordinary least-squares.

Moving-window and exponential-forgetting adaptation can be readily applied to any data-driven model by simply adapting the calibration data and using any standard training algorithm. However, this approach would require to re-train the model from scratch. Adaptation strategies can also be incorporated directly into the model training workflows to devise efficient recursive algorithms, thus taking full advantage of the computations already performed in previous training cycles. For example, updating and downdating equations have been derived for the mean vector and the covariance matrix (Wang et al., 2005), which can be used to include new observations and remove old ones in PCA and PLS, thus implementing the moving-window scheme efficiently. A recursive, exponentially-weighted PLS algorithm (based on the exponential-forgetting strategy) has been proposed by Dayal and MacGregor (1997), while Qin (1998) proposed a recursive PLS algorithm which can implement both the moving-window and the exponential-forgetting strategies. We refer the reader to the cited publications for mathematical details on the methods.

3. Model adaptation by bootstrap resampling

In the previous Section, we mentioned that the moving-window and exponential-forgetting strategies are model-agnostic, i.e., they are independent of the data-driven model used since they operate at the data level. These traditional adaptation strategies are also agnostic to the kind of change affecting the process and data (in fact, they could be run even when no change is happening at all). While this is an attractive feature for simplicity of implementation, it comes with the risk of potentially suboptimal predictive performance, particularly for abrupt changes.

Consider the case of a biomanufacturing process in which the operating volume of the main bioreactor is changed to increase the amount of the therapeutic being produced. Variations in bioreactor working volume may induce complex changes in mixing and mass transfer, which can ultimately impact biological metabolism (Facco et al., 2020). Such changes can subsequently change the correlation among process variables and the relationship between CCPs and CQAs. If a data-driven model, e.g., PLS regression, is used to predict CQAs as a function of CCPs, the volume variation can negatively impact the predictive performance of the model. The reliability of the model predictions can be checked using the T^2 and Q statistics mentioned in Section 2.1, or by computing the prediction residuals once the CQAs in the new process conditions become available. If any of these indicators signal a degradation in the predictive performance of the model, adaptation is needed.

Change-agnostic adaptation of the data-driven model will eventually restore the predictive performance of the model. However, traditional model-adaptation methods tend to be slow and perform poorly right after the change. This is due to the strong imbalance of observations of the old and new process conditions (Chu et al., 2021). However, some step changes are known in advance, such as those intentionally induced in the process as part of change controls, e.g., the aforementioned volume change. If the model diagnostics highlight the need for adaptation, the knowledge of the step change can be exploited to improve the performance of the adaptation

strategy. We propose a method to do so in this Section.

In essence, our method is based on the synthetic data augmentation strategy frequently used to tackle imbalanced learning problems (Krawczyk, 2016), namely on generating synthetic samples of the new conditions after the step change. We do so by adopting a bootstrap resampling philosophy (Efron, 1979; Efron and Tibshirani, 1993). First, we resample with replacement data from the new conditions to balance the number of observations of new and old conditions. Then, we corrupt the resampled observations with noise conforming to the noise structure of the old conditions. Finally, we gather observations of the old conditions, observations of the new conditions, and synthetic observations of the new conditions in a single dataset for model training. Repeating this procedure for multiple bootstrap-resampled datasets enables our method to obtain robust predictions and uncertainty estimation. Below, we provide additional details on the proposed approach.

Assume that a dataset $\mathcal{D}_{\text{old}}$ containing N_{old} observations of the old conditions and a dataset $\mathcal{D}_{\text{new}}$ gathering $N_{\text{new}} \ll N_{\text{old}}$ observations of the new conditions are available. A new dataset $\mathcal{D}_{\text{syn}}$ is generated by sampling with replacement $N_{\text{syn}} = N_{\text{old}} - N_{\text{new}}$ observations from $\mathcal{D}_{\text{new}}$. These observations are to be corrupted to avoid duplicates in the augmented data. A PLS model is trained on $\mathcal{D}_{\text{old}}$. Such model should be "accurate enough", i.e., yield good predictive performance of the outputs (CQAs) and capture a reasonable percentage of variance of the inputs (CCPs). The model is applied to $\mathcal{D}_{\text{old}}$ to obtain reconstruction residuals of the inputs, $\tilde{\mathbf{X}}_{\text{old}}$, and prediction residuals of the outputs, $\mathbf{H}_{\text{old}}$. The covariance matrices of the residuals, $\mathbf{S}_{X,\text{old}}$ and $\mathbf{S}_{Y,\text{old}}$, are used as estimates of the noise structure of the old data. The resampled observations in $\mathcal{D}_{\text{syn}}$ are corrupted with Gaussian noise drawn from multivariate normal distributions with null mean vectors and covariance matrices $\mathbf{S}_{X,\text{old}}$ and $\mathbf{S}_{Y,\text{old}}$. Finally, all datasets are merged as $\mathcal{D} = \mathcal{D}_{\text{old}} \cup \mathcal{D}_{\text{new}} \cup \mathcal{D}_{\text{syn}}$, and $\mathcal{D}$ is used for PLS model training.

The proposed approach is triggered after a step change degrades the predictive performance of the model, and can be repeated as observations of the new conditions are acquired and added to $\mathcal{D}_{\text{new}}$. However, note that the approach is not meant to run continuously, but simply to ease the transition between old and new process conditions. Once a sufficient number of new observations has been collected, model adaptation is deemed complete and a new PLS model can be developed on $\mathcal{D}_{\text{new}}$ only.

Given the stochastic nature of bootstrap resampling, the dataset $\mathcal{D}$ has an inherent variability, which is propagated to the PLS model and, in turn, to the CQA prediction. However, this very same variability can be used to estimate the prediction uncertainty due to the resampling procedure. The resampling from $\mathcal{D}_{\text{new}}$ is repeated R times to obtain a sequence of augmented datasets $\mathcal{D}_{\text{syn},r}$ and a corresponding sequence of predicted CQA $\hat{\mathbf{y}}_r \in \mathbb{R}^{V_Y}$, with $r \in \{1,\dots,R\}$. A robust prediction can be obtained by computing the average

$$\hat{\mathbf{y}} = \frac{1}{R}\sum_{r=1}^{R} \hat{\mathbf{y}}_r. \tag{6}$$

A confidence limits at $\alpha \in [0,1]$ significance level can be obtained by assuming that $\hat{\mathbf{y}}_r$ is normally distributed, therefore:

$$\mathrm{CL}(\hat{\mathbf{y}}) = \hat{\mathbf{y}} \pm \mathbf{s}_{\hat{\mathbf{y}}} t\,(N-A)|_{\frac{\alpha}{2}} \tag{7}$$

where $\mathbf{s}_{\hat{\mathbf{y}}}$ is the standard deviation of $\hat{\mathbf{y}}$ adjusted for the degrees of freedom of the PLS model

$$\mathbf{s}_{\hat{\mathbf{y}}} = \sqrt{\frac{1}{R-A}\sum_{r=1}^{R}\left[(\hat{\mathbf{y}}_r - \hat{\mathbf{y}})^2\right]} \tag{8}$$

and $t\,(N-A)|_{\frac{\alpha}{2}}$ is the value of a Student's t variable with $N-A$ degrees of freedom evaluated at $\frac{\alpha}{2}$ probability. The number of degrees of freedom is set to the number of latent variables A in Eq. (7) and (8). More advanced approaches to estimation of degrees of freedom in PLS model could can be used instead (Van Der Voet, 1999; Krämer and Sugiyama, 2011). We remark that the uncertainty estimation procedure is computationally intensive, therefore it is not recommended in time-sensitive applications. However, given the typically slow dynamics of biomanufacturing processes, the computational cost is not an immediate concern of the present study.

In order to run the proposed approach, some degrees of freedom need to be specified. The number of observations in the old dataset, N_{old}, can be set to include all observations of past conditions. However, in some cases it might be more practical to use just a portion of the most recent observations. This approach is motivated by the fact that the number of resampled observations, N_{syn}, should ideally be set to match $N_{\mathrm{old}} - N_{\mathrm{new}}$, thus obtaining a balanced dataset D. If N_{old} is too large, the information in D_{old} could "distract" the PLS model from the information in $D_{\mathrm{new}} \cup D_{\mathrm{syn}}$, which is essentially limited by the information in D_{new} due to the bootstrap resampling process. Furthermore, the end-of-adaptation point needs to be decided. To this end, we recommend also training a PLS model on solely on D_{new}, updating the model at each new observation. Initially, the model will probably yield poor performance and show high instability as new data are acquired, but predictive performance will eventually stabilize, at which point adaptation can be deemed complete. Finally, the number of repetitions of the bootstrap resampling, R, is to be set, if uncertainty estimation is desired. This should be large enough to ensure the statistical reliability of the sample mean $\hat{\mathbf{y}}$ and standard deviation $\mathbf{s}_{\hat{\mathbf{y}}}$.

In conclusions, we highlight the fact that the proposed approach operates at the data level, i.e., we act only on the dataset prior to training of the PLS model. Therefore, the approach can be used with any PLS training algorithm and can be applied extension of PLS as well. For example, dynamic PLS regression can be implemented by applying standard PLS to lag-augmented data matrices (Ricker, 1988), thus the resampling step in the proposed approach can be applied to the transformed matrix.

4. Case studies

In this Section, we demonstrate the proposed approach on three case studies. We first consider a simple numerical case study. We then use the IndPenSim model (Goldrick et al., 2015) to simulate a process scale-up scenario. Finally, we apply the proposed method to an industrial biomanufacturing process (Hong et al., 2023; Mohr et al., 2024). The first two case studies are developed in MatLab R2024a using the PLS Model Inversion Package code (Arnese-Feffin et al., 2025) and some additional in-house developed functions. The last case study is developed in Python using in-house developed code. We provide code and data to reproduce the first two case studies (see Data and Code availability). We do not provide data for the third case study due to confidentiality reasons.

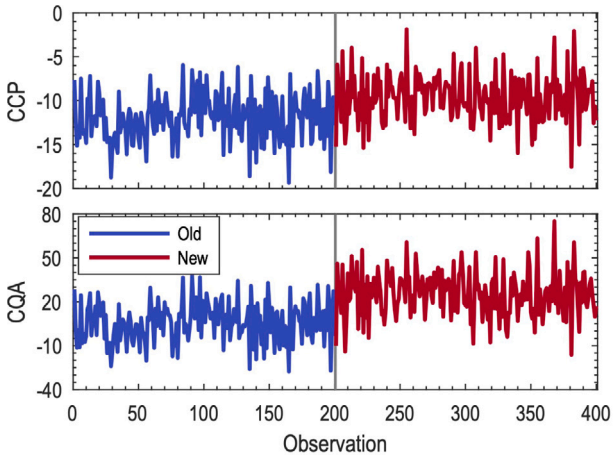

Fig. 1. Numerical case study. Selected CCP and CQA over the training dataset. The vertical line marks the onset of the step change from old to new conditions.

4.1. Numerical example

We consider a simple numerical case study first. Matrices $\mathbf{X}$ and $\mathbf{Y}$ ($N = 400$, $V_X = 12$, $V_Y = 1$) are generated based on a specified latent structure with $A = 3$ latent variables (García-Muñoz, 2004). Each variable is corrupted with Gaussian noise with a given signal-to-noise ratio to simulate measurement noise, then shifted and scaled to a randomly generated mean and standard deviation. The signal-to-noise ratios, means, and standard deviations are kept constant for the first 200 observations, then they are re-generated for the remaining 200 observations. This simulates a step change in the data affecting the means, standard deviations, and noise structures of the data, but not the underlying latent structure. These 400 observations are used training dataset, with a step-change taking place a observation 201. Furthermore, we generate $N_{\mathrm{test}} = 1000$ additional observations matching the conditions after the step change, and use these observations to test the adaptation performance of the model.

Fig. 1 shows the training data for a selected input variable (CCP) and the output variable (CQA). The effect of the step change is clearly visible. To investigate the effect of the change on the predictive performance of the model, we develop a PLS model on the first 200 observations (i.e., the old conditions) and evaluate it on the next 200 observations (i.e., the new conditions). The PLS model is calibrated with fixed $A = 3$, determined by cross-validation with one-standard-error-rule (Hastie et al., 2009). Fig. 2 shows the monitoring of T^2 and Q statistics across all 400 observations. The Q statistic is well beyond its control limit for all new observations, therefore the model predictions are unreliable after the step change, hence the need for adaptation.

We consider three adaptation strategies: moving window-adaptation (Wang et al., 2005) with $k = 30$; exponential-forgetting adaptation (Dayal and MacGregor, 1997) with $\lambda = 0.975$; and the proposed bootstrap adaptation method. As a reference to a standard industrial practice, we also consider the naïve approach of model evolution (Ramaker et al., 2005), namely the evolving model, where the model is re-trained on the entire available dataset at each new observation with no further instance weighting.

We assume to acquire the observations in the training dataset sequentially and to be aware of the step change at observation 201. We train the first PLS model after 30 observations and adapt it at each new observation thereafter according to each of the considered strategies. For the bootstrap adaptation, the model is simply evolved before the step change (i.e., adapted by including the most recent observation). After the step change, we estimate the uncertainty due to bootstrap

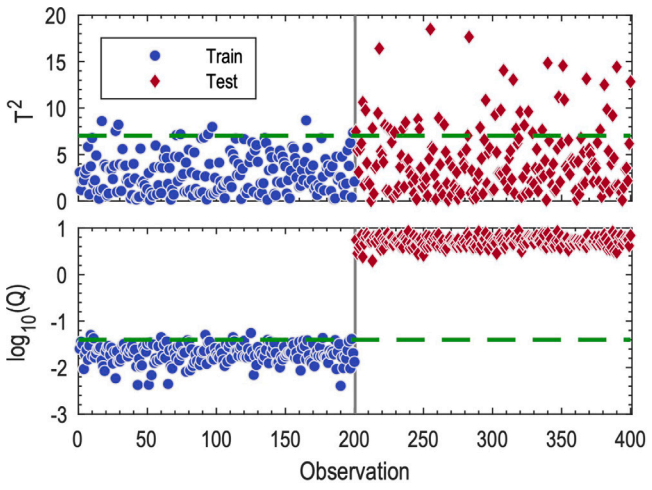

Fig. 2. Numerical case study. Monitoring statistics of a PLS model trained on data before the step change and evaluated on data after the step change. The horizontal lines are the control limits of the PLS monitoring statistics. The vertical line marks the onset of the step change from old to new conditions.

resampling by performing $R = 100$ repetitions and computing the confidence limit at $\alpha = 0.05$ significance level.

Model evolution, moving window-adaptation, and bootstrap adaptation are implemented by re-training the PLS model each time that the dataset is updated. At each re-training, we preprocess the data matrices by autoscaling (i.e., each column is normalized to zero mean and unit variance); we use the means and variances of the training matrices to preprocess new data in the model evaluation phase. Exponential-forgetting adaptation is carried out using a specialized algorithm and preprocessing strategy (Dayal and MacGregor, 1997). The number of latent variables is set as $A = 3$, determined by cross-validation with a one-standard-error rule on the training dataset portion prior to the step change. We have considered the case where A is determined by cross-validation each time a PLS model is trained. However, we did not find any significant difference from the case where A is kept fixed, therefore we chose to fix A for simplicity.

We measure the performance of each adaptation method by monitoring the RMSE on the independent test dataset (which is consistent with the conditions after the step change). After a new observation is acquired, we adapt the model, apply it to the entire testing dataset, and compute the RMSE of the predictions. We are interested in assessing how quickly the testing RMSE approaches an "optimal" value after the step change, i.e., how quickly the considered methods adapt the model to new conditions. We define a reference error by training a PLS model only on the training observations after the step change and computing the RMSE on the test dataset. The testing RMSEs of the considered adaptation methods are reported in Fig. 3, where Fig. 3(b) shows an enlargement of the region after the step change in Fig. 3(a).

The proposed approach achieves a predictive performance very close to the reference after just 2 observations are acquired. On the other hand, $\sim$ 12 observations are needed in the moving-window scheme, while the exponential-forgetting strategy requires > 50 observations in the considered case study (even more than simple model evolution). The bootstrap approach also shows more stable model performance before and after the step change as all available observations are always used, unlike in traditional adaptation schemes. However, we also point out that our approach does not converge to the reference performance unless the adaptation is deemed complete and a new PLS model is trained on the new observations alone. This shows that the bootstrap-based adaptation is tailored to handle the transition period right after the step change, rather than being run continuously as the other strategies.

Table 1
List of features generated from time profiles of batches in the IndPenSim case study.

Feature name	Symbol	Units
Input variables		
Batch duration	t	h
Harvested volume	V_{tot}	m^3
Substrate mass fed	$m_{\text{S}}{}^{\text{a}}$	kg
Substrate mass consumed	$m_{\text{S}}{}^{\text{c}}$	kg
Anti-foam mass fed	$m_{\text{oil}}{}^{\text{a}}$	kg
Phenylacetic acid mass fed	$m_{\text{PAA}}{}^{\text{a}}$	kg
Acid mass added	$m_{\text{a}}{}^{\text{a}}$	kg
Base mass added	$m_{\text{b}}{}^{\text{a}}$	kg
Heat removed	Q_{c}	MJ
Heat provided	Q_{h}	MJ
Dilution water mass fed	$m_{\text{W}}{}^{\text{a}}$	kg
Average temperature	T_{avg}	K
Average pressure	P_{avg}	bar
Average pH	$\text{p}H_{\text{avg}}$	
Average dissolved oxygen	DO_{avg}	%
Average oxygen uptake rate	OUR_{avg}	g h^{-1}
Average carbon evolution rate	CER_{avg}	g h^{-1}
Final biomass concentration	X	g L^{-1}
Final substrate concentration	c_{S}	g L^{-1}
Final ammonia concentration	$c\text{NH}_3$	g L^{-1}
Final phenylacetic acid concentration	c_{PAA}	g L^{-1}
Final viscosity	μ	cP
Output variables		
Final penicillin concentration	c_{P}	g L^{-1}

4.2. Penicillin production process

We use IndPenSim (Goldrick et al., 2015) to test the proposed adaptation strategy in a realistic scenario. The simulator implements a model for a fed-batch, industrial-scale penicillin production process. Time profiles of variables returned by the simulator are first corrupted with appropriate Gaussian noise to simulate real measurements, then used to compute features characterizing single batch runs, which are regarded as observations in the dataset. A list of the features used in this case study is reported in Table 1, with their function in the PLS model. The purpose of the soft-sensor is to estimate the end-of-batch penicillin concentration.

We simulate a scale-up scenario as the one described in Section 3: the variation of the volume of the bioreactor. 100 batches are generated by scaling the initial volume of the batch to one half of the nominal reactor capacity defined by the simulator. Profiles of recipe-driven variables (i.e., the feed flow-rates not manipulated by the control system) are scaled and modified to maintain good process performance. Afterwards, we generate 100 more batches operating the reactor at full capacity. These 200 batches are regarded as the training dataset for the PLS model. Finally, we generate 100 additional full-capacity batches to be used as a testing dataset. Selected features for the training dataset are reported in Fig. 4.

Similarly to the previous case study, the monitoring statistics of a PLS model ($A = 7$ latent variables, determined by cross-validation) calibrated on the old conditions and evaluated on the new conditions highlight the need for model adaptation. The statistics are shown in Fig. 5.

Similarly to the previous case study, we compare the performance of moving-window ($k = 40$), exponential-forgetting ($\lambda = 0.995$), bootstrap adaptation, and model evolution. We again acquire observations sequentially and train the first model after 40 observations. Model training, adaptation, and performance evaluation are done as described in Section 4.1. The testing RMSEs are reported in Fig. 6.

While the spread of performance of different methods is not as wide as in the previous case study, the proposed approach still achieves the best performance after just 1 batch in the conditions: the testing RMSE of the PLS model adapted using the bootstrap is $\sim$ 15% lower

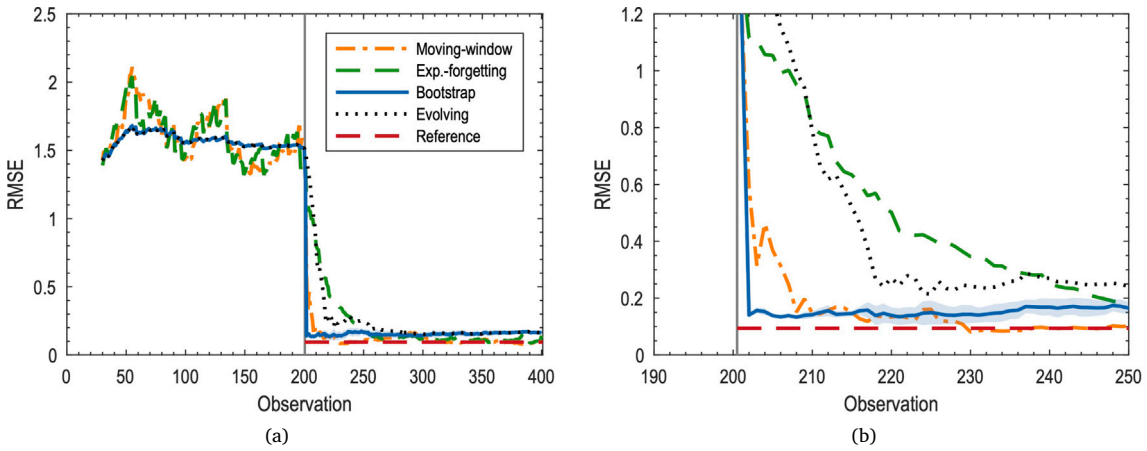

Fig. 3. Numerical case study. Comparison of performance of the adaptation methods. The RMSE at observation i is obtained by applying the PLS model trained at observation i (based on the relevant adaptation strategy) to the test dataset. The envelope around the bootstrap curve is the confidence limit with $\alpha = 0.05$. The horizontal dotted line is the reference error. The vertical line indicates the step change in the data.

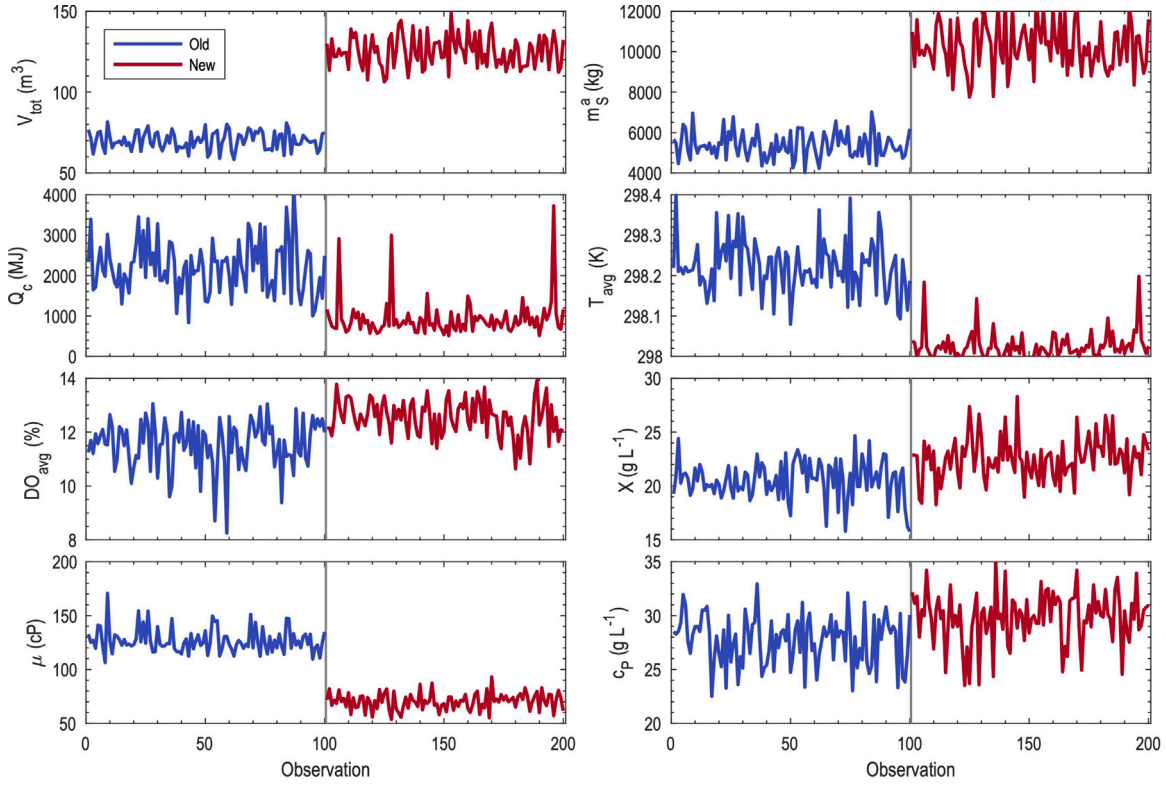

Fig. 4. IndPenSim case study. Selected features over the training dataset. The end-of-batch penicillin concentration (c_P) is the CQA. The vertical line marks the onset of the step change from old to new conditions.

than the one achieved by the moving-window scheme, and $\sim$ 10% lower than the exponential-forgetting scheme. As in the previous case study, our approach yields models with more stable performance in the periods before and after the step change due to an optimal use of all observations available. Fig. 6(a) also shows that the RMSE of the model adapted using the proposed approach converges to the reference similarly to the models yielded by the other adaptation strategies.

From Fig. 6b, we see a high test RMSE at observation 105 for our bootstrap method. This can be explained by sharp peaks on Q_c and T_{avg} at observation 105 in Fig. 4. While these single peaks do not have a strong impact on the PLS model constructed by the moving-window and the exponential-forgetting methods, the peaks affect $\sim$ 20% of $\mathcal{D}_{syn}$ (i.e., $\sim$ 10% of the balanced dataset) for the bootstrap method since every measurement in $\mathcal{D}_{new}$ serves as a baseline for the bootstrap resampling. One possible way to improve this limitation is to investigate whether these outliers are due to sensor error using the domain knowledge, and replace them with reasonable values based on the latent relationship among the variables (Rhyu et al., 2024).

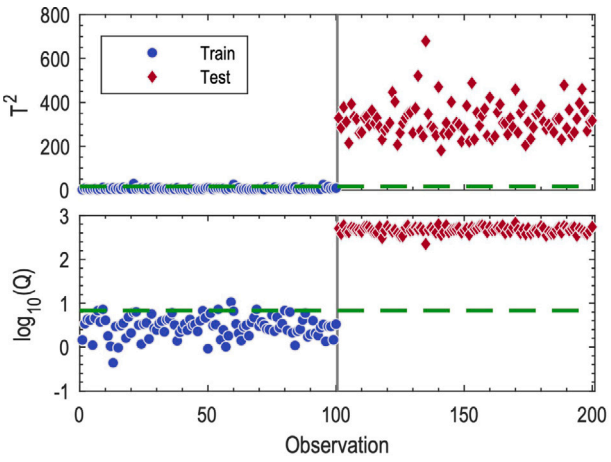

Fig. 5. IndPenSim case study. Monitoring statistics of a PLS model trained on data before the step change and evaluated on data after the step change. The horizontal lines are the control limits of the PLS monitoring statistics. The vertical line marks the onset of the step change from old to new conditions.

Table 2
List of campaigns used in the industrial case study. The values greater than 7.5 are bolded.

Campaign	Observation	CQA range	Campaign	Observation	CQA range
A	1–4	6.54–**8.15**	G	37–53	6.07–7.20
B	5–8	5.84–6.64	H	54–56	6.73–7.06
C	9–12	6.55–6.83	I	57–59	**7.99–8.14**
D	13–20	6.15–7.00	J	60–70	**7.59–8.95**
E	21–23	6.76–7.07	K	71–77	6.82–**8.09**
F	24–36	6.15–7.06			

4.3. End-to-end biomanufacturing process

In this Section, we evaluate our model adaptation method by bootstrap resampling on an industrial case study (Hong et al., 2023; Mohr et al., 2024). The dataset was obtained from a monoclonal antibody production process consisting of cell-growth expansion steps, antibody production, product harvest, and subsequent purification. The dataset captures batch-to-batch dynamics across 77 batches (i.e., observations) from 11 manufacturing campaigns, each reporting 169 input variables from the bioreactors, harvest, columns, and viral inactivation operations. The CQA is the antibody's basic peaks percentage measured at column 2 (Hong et al., 2023; Mohr et al., 2024). Among 169 measured variables, the harvest feed flow rate was not considered in this study since the value remained constant across all observations except for two batches with a negligible deviation. This results in matrices $\mathbf{X}$ and $\mathbf{Y}$ with $N = 77$, $V_X = 168$, and $V_Y = 1$. Details of each campaign are listed in Table 2.

We use the observations from the last campaign (i.e., campaign K) as the test set, while treating all other observations (i.e., campaigns A–J) as the training set. To determine when the main step change has occurred, we first determine A for PCA using 10 repetitions of 5-fold cross validation after each campaign transition. Then, we calculate T^2 and Q statistics for the observation i using observations 1 to $(i-1)$. The control limit for the T^2 statistic is calculated by setting the one-tail confidence limit of $\alpha = 0.05$ with the assumption that the T^2 statistic follows the χ^2-distribution. From Fig. 7a, we conclude that the step change occurred at the campaign transition H→I since the T^2 statistic exceeds the threshold two times in a row. This conclusion matches with the previous study in Mohr et al. (2024), and with Fig. 7b that shows the PLS plot of all observations with $A = 2$, where A was determined from the same method as in the PCA case.

Unlike the previous case studies, the dataset in the industrial case study is composed of multiple step changes, as described in Appendix A.1 . Therefore, CQA prediction of the test set (i.e., campaign K) with observations only after the main step change (i.e., campaigns I–J) performs poorly, especially due to a small number of samples (i.e., 14).

Table 3
Performance of the adaptation methods for the industrial case study. Bootstrap performance is the average value across $R = 100$.

Observation	Evolving	Moving-window	Exp.-forgetting	Bootstrap
56	1.43	1.26	0.90	–
57	1.12	0.73	0.57	0.68
58	0.78	0.49	0.50	0.48

To accommodate this issue, the reference model in this Section is defined as the PLS model trained with all 70 observations in the training set.

Here, we compare the performance of moving-window ($k = 20$), exponential-forgetting ($\lambda = 0.90$), and bootstrap adaptation triggered at the campaign transition H→I. The schematics of the evolving model, as well as three adaptation models, are demonstrated in Fig. 8. Each subplot visualizes the PLS model constructed by each adaptation method with $A = 2$. Contours represent the PLS model trained with $\mathcal{D}_{\text{old}}$ and $\mathcal{D}_{\text{new}}$ (and $\mathcal{D}_{\text{syn}}$ for the bootstrapping method), and the coordinates of the test set is determined based on the loading matrix of the trained PLS model.

From Fig. 9 and Table 3, we see that all three adaptation methods outperform the evolving model, and they quickly converge to the reference model. However, there is no big difference across the three adaptation methods. In fact, the RMSE of the exponential forgetting method at observation 57 is ∼ 16% lower than the bootstrap method. However, there are several things that need to be considered for a fair comparison.

Firstly, there are several minor step changes going on after the main step change at campaign transition H→I. This indicates that observations 57 and 58, which are from campaign I, could not fully represent the test set, which is from campaign K. The deviation between campaigns I and K can also be observed in Fig. 7b. In fact, when changing the test set from campaign K to campaign J (i.e., training set is campaigns A–I and K), we observe in Fig. 10a and Table 4 that the exponential-forgetting method shows the worst adaptability. Since Fig. 7b indicates that the campaign J is more representative than the campaign K for the post-step change regime, the results in Fig. 9 and Table 3 cannot be interpreted as the exponential-forgetting method is the best. A more valid comparison would have been possible if there was a campaign with a sufficient number of observations, allowing them to be split into observations in the training set after the step change and observations in the test set.

Secondly, the adaptability performance of moving-window and exponential-forgetting methods highly depends on the order of observations in the training set. This dependency is suitable for capturing slow dynamics (e.g., evolutionary changes), and it could improve the adaptability performance if the direction of slow dynamics aligns with the intended abrupt change. From Fig. 7b, we observe that the campaign H serves as a "bridge" between campaigns A–G and campaigns I–K. This slow transition helps the moving-window and exponential-forgetting methods to be better prepared for the abrupt change, as can be observed in Fig. 9b where the RMSE metrics of these two methods suddenly decrease at the campaign transition G→H. However, the impact of such dependency should be removed when comparing only the adaptability performance with respect to abrupt changes. In practice, the alignment of campaign-to-campaign transitions and the intended abrupt change is not guaranteed. For the hypothetical case where the observations in campaigns A–H are observed in a reverse order, we see from Fig. 10b and Table 4 that our bootstrap method outperforms the other two adaptation methods. This result concludes that the adaptability performance of the moving-window and

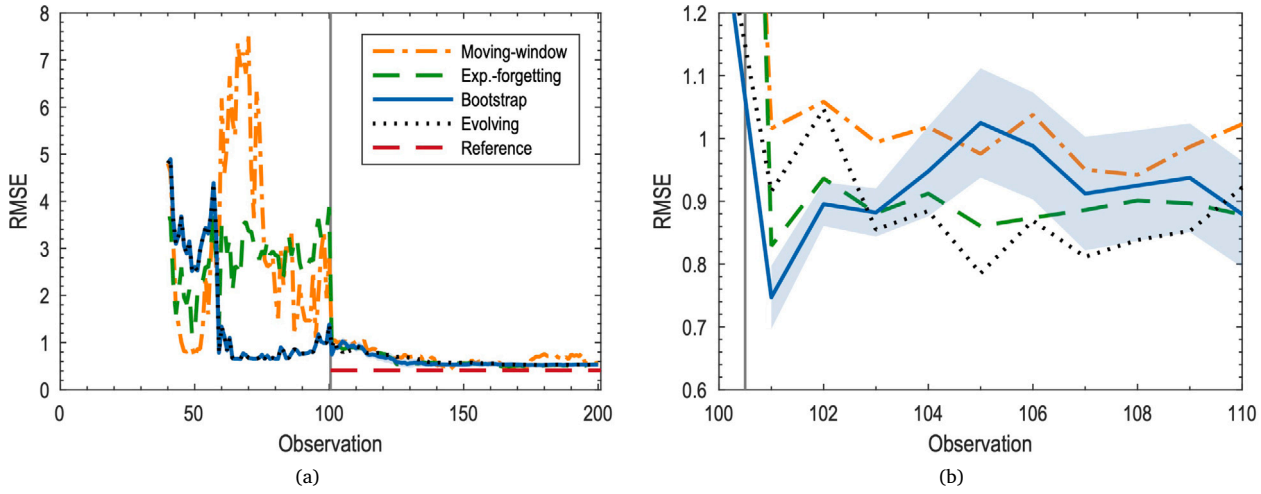

Fig. 6. IndPenSim case study. Comparison of performance of the adaptation methods. The RMSE at observation i is obtained by applying the PLS model trained at observation i (based on the relevant adaptation strategy) to the test dataset. The envelope around the bootstrap curve is the confidence limit with $\alpha = 0.05$. The horizontal dotted line is the reference error. The vertical line indicates the step change in the data.

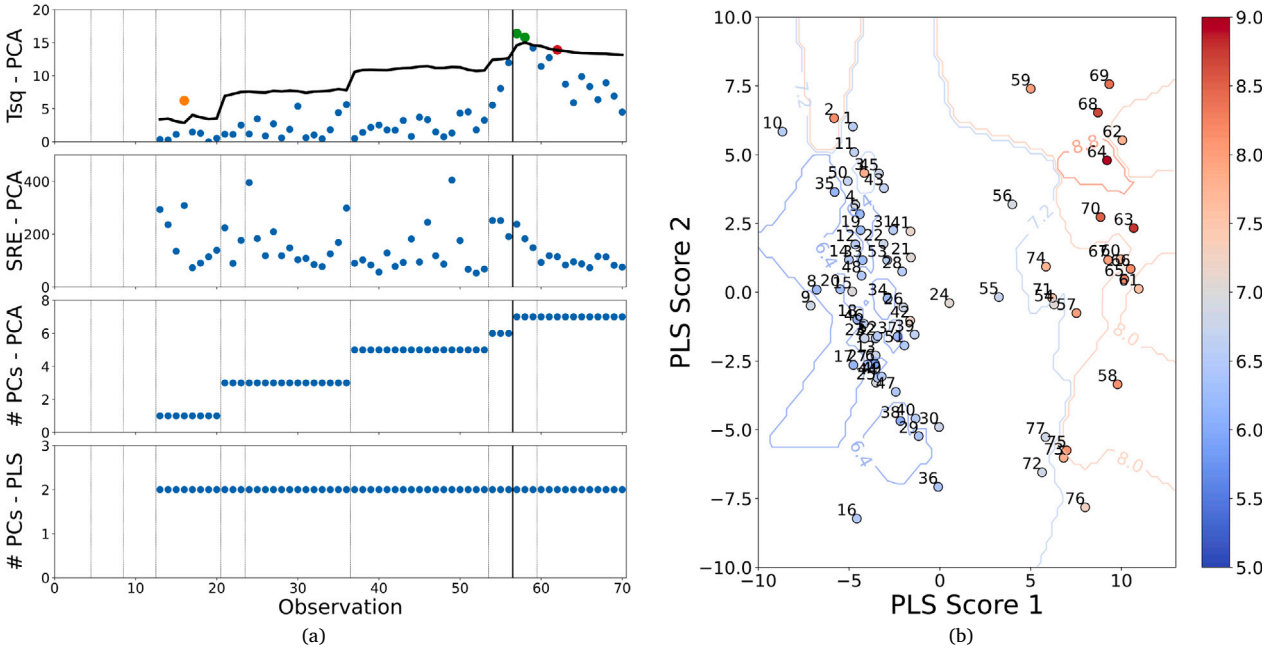

Fig. 7. Dataset used for the industrial case study. (a) T^2 and Q statistics, and A's at PCA and PLS for observation i. Observations where the T^2 statistic exceeds the control limit are marked with colors other than blue. Vertical dotted lines indicate the campaign transition. (b) PLS model of the entire dataset with $A = 2$.

Table 4
Performance of the adaptation methods for the adjusted industrial case studies.

Change test set from campaign K to campaign J				
Observation	Evolving	Moving-window	Exp.-forgetting	Bootstrap
56	1.67	1.49	1.44	–
57	1.13	0.90	0.97	0.93
58	0.82	0.75	0.88	0.69
Reverse order of observations in campaigns A–H				
Observation	Evolving	Moving-window	Exp.-forgetting	Bootstrap
56	1.43	1.45	1.66	–
57	1.12	0.91	0.65	0.68
58	0.78	0.71	0.57	0.48

the exponential-forgetting methods in Fig. 9 and Table 3 might have been overestimated.

Overall, our bootstrap method does not outperform the other two adaptation methods when applied to the industrial dataset composed of multiple step changes. However, it should be noted that (1) the observations used as the test set are from a separate campaign than the training set due to the limited number of observations per campaign, and (2) the dataset used in this case study was from the case where the direction of campaign-to-campaign transitions aligned with the intended abrupt change.

5. Conclusions

In this study, we proposed a novel model adaptation strategy to tackle step changes in the data. The knowledge of the presence of the

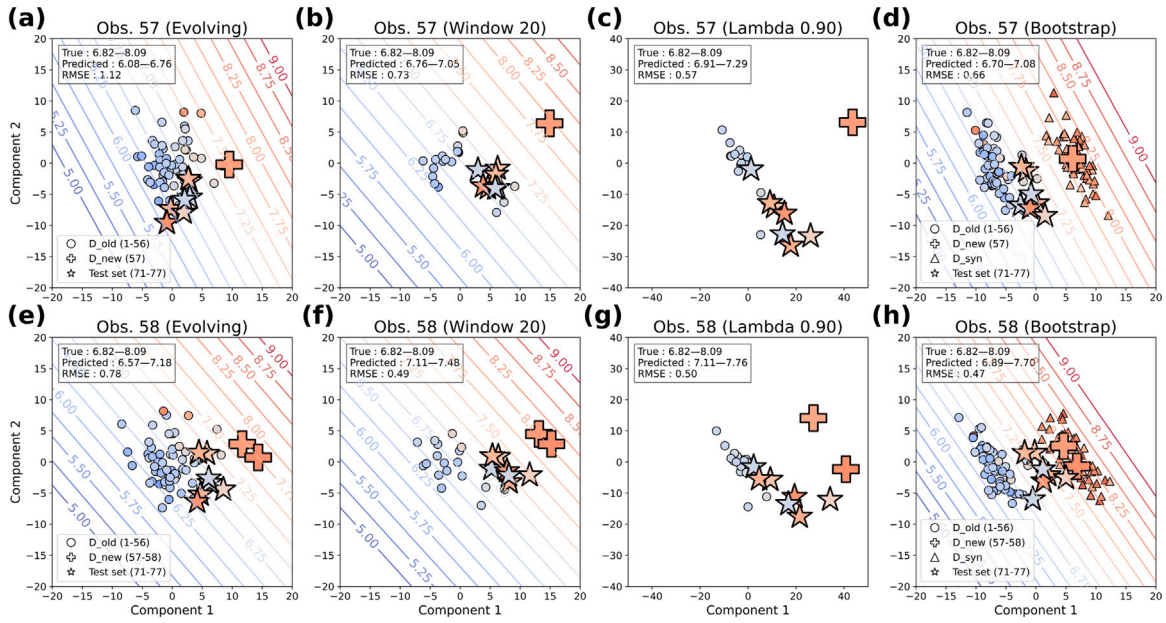

Fig. 8. Schematic on how each method adapts the PLS model. The colors indicate the measured CQA values at each observation. For the exponential-forgetting adaptation method, where the PLS model cannot be explicitly displayed in the fixed coordinate, the position of data points is set by the scores matrix that is calculated on the weighted and mean-centered training set. (a–d) Model adaptation after observation 57 using the evolving, moving-window, exponential-forgetting, and bootstrap adaptation methods, respectively. (e–h) Model adaptation after observation 58.

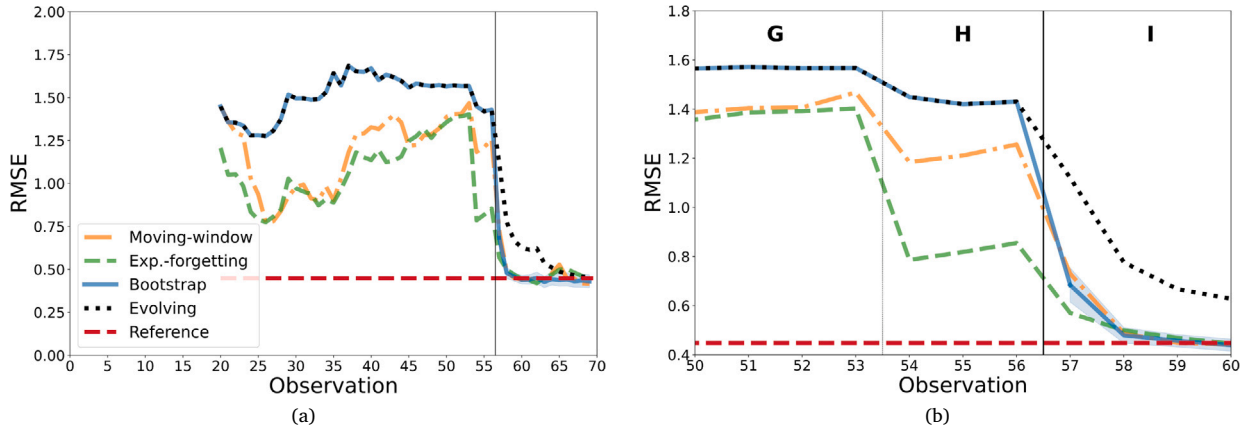

Fig. 9. Industrial case study. Comparison of performance of the adaptation methods. The RMSE at observation i is obtained by applying the PLS model trained at observation i (based on the relevant adaptation strategy) to the test dataset. The envelope around the bootstrap curve is the confidence limit with $\alpha = 0.05$. The horizontal dotted line is the reference error. Unlike the previous case studies, the horizontal dotted line spans observations before the main step change. This is because the reference model is trained by all observations in the training set. The vertical lines indicate the campaign transition where the main step change is H→I.

step change is exploited to inform the proposed method, thus using the available data in an optimal way. Specifically, we leverage the bootstrap philosophy to balance the dataset right after the step change to overcome the degradation of predictive performance of the model induced by strong imbalance in the dataset. We do so by sampling with replacement from the new data to obtain a number of observations such that the overall dataset is balanced. We then corrupt the sampled observation by adding Gaussian noise with zero mean and covariance consistent with the noise in the old data (estimated using a model of the old data alone) to obtain a realistic augmented dataset. Repeating the resampling procedure multiple times equips our approach with uncertainty estimation capabilities.

We demonstrated our approach considering soft sensing problems, where a data-driven model, namely PLS regression, is used to predict a hard-to-measure CQA based on easy-to-measure CCPs. We considered three case studies: a numerical case study, a case study based on the IndPenSim model (both set up to simulate a process scale-up scenario), and an industrial biomanufacturing process. In comparison to traditional adaptation strategies, namely moving-average and exponential-forgetting schemes, our bootstrap-based approach achieves faster adaptation, and better and more stable predictive performance, highlighting the importance of incorporating knowledge on the process change being tackled into the model adaptation procedure.

However, it was also observed that our bootstrap resampling approach could be sensitive to outliers present in the early post-step change regime. In order to make our method more robust to these outliers, we will work on combining the outlier-missingness work (Rhyu et al., 2024) with our bootstrap method to reject the outliers from (1)

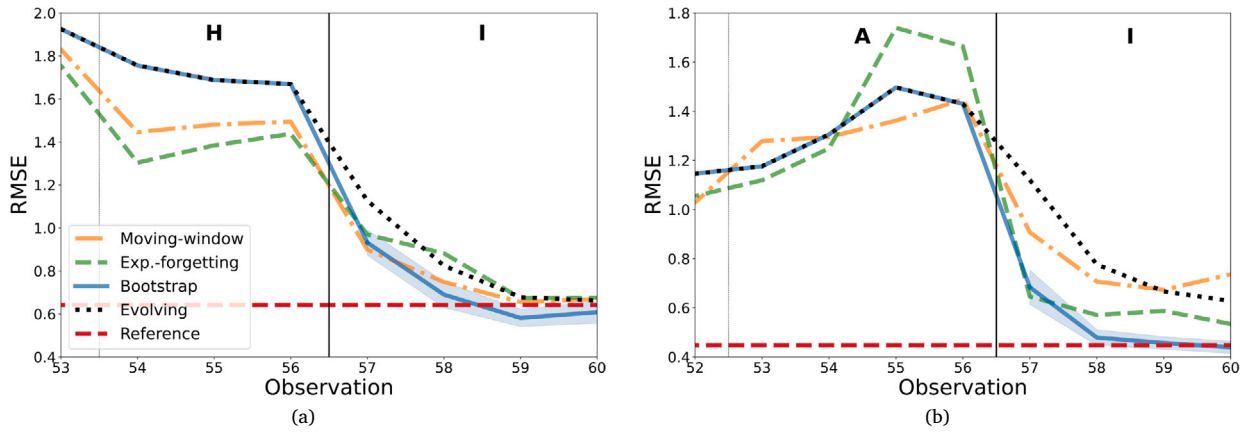

Fig. 10. Adjusted industrial case studies. (a) Case when the test set is changed from campaign K to campaign J. (b) Case when the order of observations in campaigns A–H is reversed.

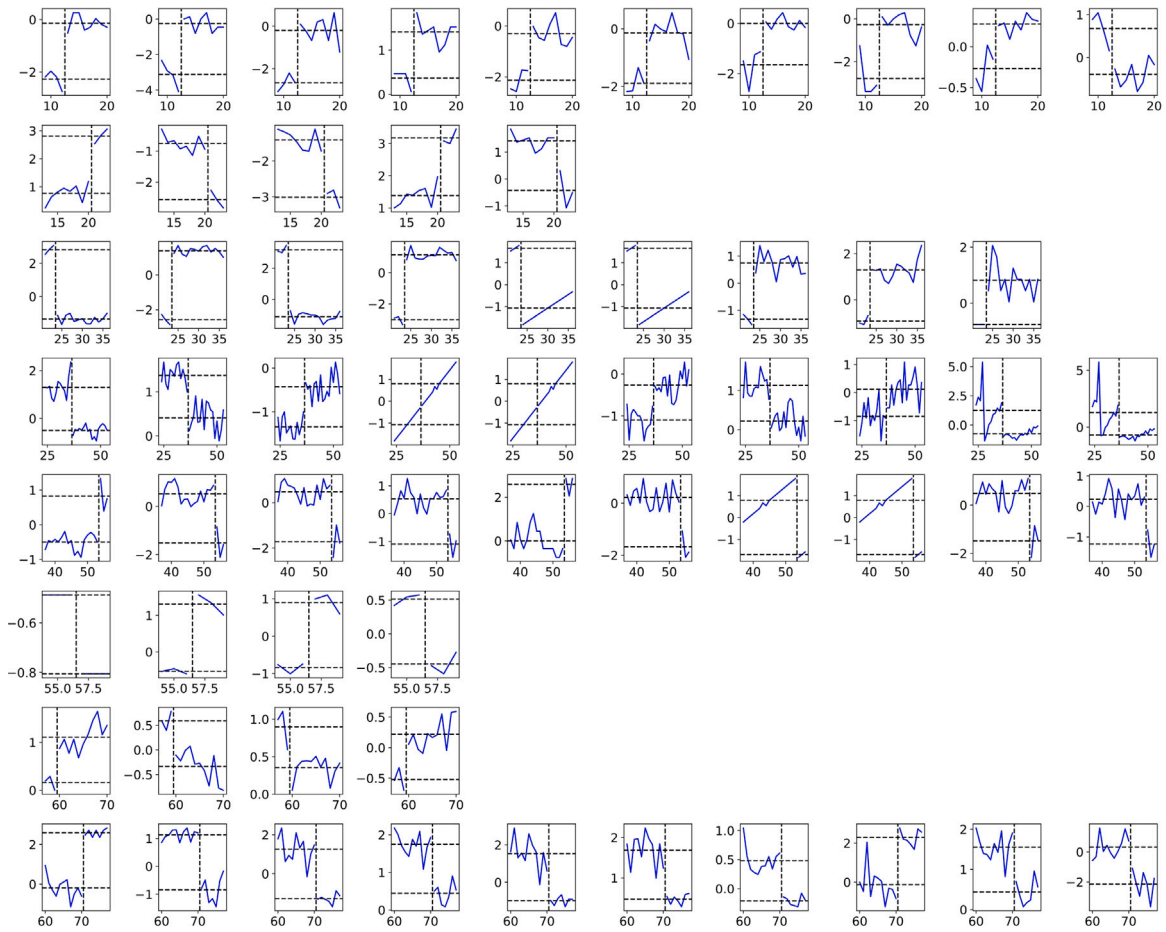

Fig. A.1. Input variables that changed abruptly during campaign transition. Each subplot shows one input variable, with the black dotted vertical line indicating where the transition occurred. Two dotted horizontal lines indicate the average of input variables measured at each campaign. Only variables with a t-test p-value smaller than 0.001 are plotted, and only the ten with the smallest p-values are shown if more than ten. Each row corresponds to a separate transition of manufacturing campaigns. Every value is standardized using the full observations.

setting unrealistic values as the baseline for the resampling, and (2) distorting the noise structure in the training set.

CRediT authorship contribution statement

Elia Arnese-Feffin: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jinwook Rhyu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Benjamin T. Smith:** Writing – review & editing, Writing – original draft, Validation, Data curation, Conceptualization. **Chris D. Castro:** Writing – review & editing, Writing – original draft, Validation, Data curation, Conceptualization. **Jacqueline M.**

Wolfrum: Writing – review & editing, Resources, Conceptualization. **Stacy L. Springs:** Writing – review & editing, Resources, Conceptualization. **Roger A. Hart:** Writing – review & editing, Resources, Funding acquisition, Conceptualization. **Tom Mistretta:** Writing – review & editing, Resources, Conceptualization. **Richard D. Braatz:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the U.S. Food and Drug Administration under the FDA BAA-22-00123 program, Award Number 75F40122C00200. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the financial sponsor.

Appendix. Details for the industrial case study

A.1. Input variable change during campaign transition

Fig. A.1 displays the representative input variables that changed abruptly between the consecutive manufacturing campaigns. We used a two-sample t-test to determine whether the mean of the input variables has changed (i.e., whether the step change has occurred) during the campaign transition. The variance of the two distributions was assumed to be the same, and the confidence limit was set to 0.001 for plotting Fig. A.1. We observe from Fig. A.1 that at least four input variables experienced a notable shift at each campaign transition, indicating that every transition can be considered a step change.

Data availability

The code used to obtain the results discussed in this article and data for the case studies discussed in Section 4.1 and Section 4.2 are available at the following GitHub repository: https://github.com/EliaAF/PLSBootstrapAdaptation. If the manuscript is accepted for publication, the code will be made available to a wider audience through a public GitHub repository. The data used in the case study presented in Section 4.3 are confidential.

References

Arnese-Feffin, E., Facco, P., Bezzo, F., Barolo, M., 2025. Systematizing product design by latent-variable modeling – a unifying framework for the formulation and solution of PLS model inversion problems. Chem. Eng. Sci. 299, 120505. http://dx.doi.org/10.1016/j.ces.2024.120505.

Briceno-Mena, L.A., Arges, C.G., Romagnoli, J.A., 2023. Machine learning-based surrogate models and transfer learning for derivative free optimization of HT-PEM fuel cells. Comput. Chem. Eng. 171, 108159. http://dx.doi.org/10.1016/j.compchemeng.2023.108159.

Bro, R., Kjeldahl, K., Smilde, A.K., Kiers, H.A.L., 2008. Cross-validation of component models: A critical look at current methods. Anal. Bioanal. Chem. 390, 1241–1251. http://dx.doi.org/10.1007/s00216-007-1790-1.

Chiang, L.H., Russel, E.L., Braatz, R.D., 2001. Fault Detection and Diagnosis in Industrial Systems, first ed. Springer.

Chu, F., Wang, J., Zhao, X., Zhang, S., Chen, T., Jia, R., Xiong, G., 2021. Transfer learning for nonlinear batch process operation optimization. J. Process Control 101, 11–23. http://dx.doi.org/10.1016/j.jprocont.2021.03.002.

Dayal, B.S., MacGregor, J.F., 1997. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. J. Process Control 7, 169–179. http://dx.doi.org/10.1016/S0959-1524(97)80001-7.

Efron, B., 1979. Bootstrap methods: Another look at the Jackknife. Ann. Statist. 7, 1–26, The Annals of Statistics.

Efron, B., Tibshirani, R., 1993. An Introduction to the Bootstrap. Chapman & Hall/CRC.

Facco, P., Zomer, S., Rowland-Jones, R.C., Marsh, D., Diaz-Fernandez, P., Finka, G., Bezzo, F., Barolo, M., 2020. Using data analytics to accelerate biopharmaceutical process scale-up. Biochem. Eng. J. 164, 107791. http://dx.doi.org/10.1016/j.bej.2020.107791.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A., 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46, 1–37. http://dx.doi.org/10.1145/2523813.

García-Muñoz, S., 2004. Batch Process Improvement using Latent Variable Methods (Ph.D. thesis). McMaster University..

Geladi, P., Kowalski, B.R., 1986. Partial least-squares regression: A tutorial. Chemometr. Intell. Lab. Syst. 185, 1–17. http://dx.doi.org/10.1016/0003-2670(86)80028-9.

Goldrick, S., Ştefan, A., Lovett, D., Montague, G., Lennox, B., 2015. The development of an industrial-scale fed-batch fermentation simulation. J. Biotech. 193, 70–82. http://dx.doi.org/10.1016/j.jbiotec.2014.10.029.

Hastie, T., Tibshirani, R., Friedman, J.H., 2009. The Elements of Statistical Learning, second ed. Springer.

Hong, M.S., Mohr, F., Castro, C.D., Smith, B.T., Wolfrum, J.M., Springs, S.L., Sinskey, A.J., Hart, R.A., Mistretta, T., Braatz, R.D., 2023. Smart process analytics for the end-to-end batch manufacturing of monoclonal antibodies. Comput. Chem. Eng. 179, 108445. http://dx.doi.org/10.1016/j.compchemeng.2023.108445.

Kadlec, P., Gabrys, B., Strandt, S., 2009. Data-driven soft sensors in the process industry. Comput. Chem. Eng. 33, 795–814. http://dx.doi.org/10.1016/j.compchemeng.2008.12.012.

Kadlec, P., Grbíc, R., Gabrys, B., 2011. Review of adaptation mechanisms for data-driven soft sensors. Comput. Chem. Eng. 35, 1–24. http://dx.doi.org/10.1016/j.compchemeng.2010.07.034.

Krämer, N., Sugiyama, M., 2011. The degrees of freedom of partial least squares regression. J. Amer. Statist. Assoc. 106, 697–705. http://dx.doi.org/10.1198/jasa.2011.tm10107.

Krawczyk, B., 2016. Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. 5, 221–232. http://dx.doi.org/10.1007/s13748-016-0094-0.

Lima, M., Neto, M., Filho, T.S., De A. Fagundes, R.A., 2022. Learning under concept drift for regression–a systematic literature review. IEEE Access 10, 45410–45429. http://dx.doi.org/10.1109/ACCESS.2022.3169785.

Louwerse, D.J., Smilde, A.K., Kiers, H.A.L., 1999. Cross-validation of multiway component models. J. Chemom. 13, 491–510. http://dx.doi.org/10.1002/(SICI)1099-128X(199909/10)13:5<491::AID-CEM537>3.0.CO;2-2.

Mohr, F., Arnese-Feffin, E., Barolo, M., Braatz, R.D., 2025. Smart process analytics for process monitoring. Comput. Chem. Eng. 194, 108918. http://dx.doi.org/10.1016/j.compchemeng.2024.108918.

Mohr, F., Hong, M.S., Castro, C.D., Smith, B.T., Wolfrum, J.M., Springs, S.L., Sinskey, A.J., Hart, R.A., Mistretta, T., Braatz, R.D., 2024. Tensorial approaches combining time series and batch data for the end-to-end batch manufacturing of monoclonal antibodies. Comput. Chem. Eng. 182, 108557. http://dx.doi.org/10.1016/j.compchemeng.2023.108557.

O'Flaherty, R., Bergin, A., Flampouri, E., Martins Mota, L., Obaidi, I., Quigley, A., Xie, Y., Butler, M., 2020. Mammalian cell culture for production of recombinant proteins: A review of the critical steps in their biomanufacturing. Biotech. Adv. 43, 107552. http://dx.doi.org/10.1016/j.biotechadv.2020.107552.

Qin, S.J., 1998. Recursive PLS algorithms for adaptive data modeling. Comput. Chem. Eng. 22, 503–514. http://dx.doi.org/10.1016/s0098-1354(97)00262-7.

Qin, S.J., 2003. Statistical process monitoring: Basics and beyond. J. Chemom. 17, 480–502. http://dx.doi.org/10.1002/cem.800.

Ramaker, H.J., van Sprang, E.N., Westerhuis, J.A., Smilde, A.K., 2005. Fault detection properties of global, local and time evolving models for batch process monitoring. J. Process Control 15, 799–805. http://dx.doi.org/10.1016/j.jprocont.2005.02.001.

Reis, M.S., Gins, G., 2017. Industrial process monitoring in the big data/industry 4.0 era: From detection, to diagnosis, to prognosis. Processes 5, 35. http://dx.doi.org/10.3390/pr5030035.

Rhyu, J., Bozinovski, D., Dubs, A.B., Mohan, N., Cummings Bende, E.M., Maloney, A.J., Nieves, M., Sangerman, J., Lu, A.E., Hong, M.S., Artamonova, A., Ou, R.W., Barone, P.W., Leung, J.C., Wolfrum, J.M., Sinskey, A.J., Springs, S.L., Braatz, R.D., 2024. Automated outlier detection and estimation of missing data. Comput. Chem. Eng. 180, 108448. http://dx.doi.org/10.1016/j.compchemeng.2023.108448.

Ricker, N., 1988. The use of biased least-squares estimators for parameters in discrete-time pulse-response models. Ind. Eng. Chem. Res. 27, 343–350. http://dx.doi.org/10.1021/ie00074a023.

Ündey, C., Tatara, E., Çınar, A., 2004. Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations. J. Biotech. 108, 61–77. http://dx.doi.org/10.1016/j.jbiotec.2003.10.004.

Van Der Voet, H., 1999. Pseudo-degrees of freedom for complex predictive models: The example of partial least squares. J. Chemom. 13, 195–208. http://dx.doi.org/10.1002/(SICI)1099-128X(199905/08)13:3/4<195::AID-CEM540>3.0.CO;2-L.

Wang, X., Kruger, U., Irwin, W., 2005. Process monitoring approach using fast moving window PCA. Ind. Eng. Chem. Res. 44, 5691–5702. http://dx.doi.org/10.1021/ie048873f.

Wise, B.M., Gallagher, N.B., 1996. The process chemometrics approach to process monitoring and fault detection. J. Process Control 6, 329–348. http://dx.doi.org/10.1016/0959-1524(96)00009-1.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal components analysis. Chemometr. Intell. Lab. Syst. 2, 37–52. http://dx.doi.org/10.1016/0169-7439(87)80084-9.

Wold, S., Sjöström, M., Eriksson, L., 2001. PLS-regression: A basic tool of chemometrics. Chemometr. Intell. Lab. Syst. 58, 109–130. http://dx.doi.org/10.1016/S0169-7439(01)00155-1.

Zhu, X., Rehman, K.U., Wang, B., Shahzad, M., 2020. Modern soft-sensing modeling methods for fermentation processes. Sensors 20, 1771. http://dx.doi.org/10.3390/s20061771.