Continual learning for multimode dynamic process monitoring with applications to an ultra–supercritical thermal power plant

Article

Accepted Version

Continual learning for multimode dynamic process monitoring with applications to an ultra–supercritical thermal power plant

Jingxin Zhang, Donghua Zhou, *Fellow, IEEE*, Maoyin Chen, and Xia Hong, *Senior Member, IEEE*

***Abstract*—This paper introduces a novel sparse dynamic inner principal component analysis (SDiPCA) based monitoring for multimode dynamic processes. Different from traditional multimode monitoring algorithms, a model is updated for sequential modes by memorizing the significant features of existing modes. By adopting the concept of intelligent synapses in continual learning, a loss of quadratic term is introduced to penalize the changes of mode–relevant parameters, where modified synaptic intelligence (MSI) is proposed to estimate the parameter importance. Thus, the proposed algorithm is referred to as SDiPCA–MSI. When a new mode arrives, a set of normal samples should be collected. The previous significant features are consolidated without explicitly storing training samples, while extracting new information from the current mode. Consequently, SDiPCA–MSI can provide outstanding performance for successive modes. Characteristics of the proposed approach are discussed, including the computational complexity, advantages and potential limitations. Compared with several state-of-the-art monitoring methods, the effectiveness and superiorities of the proposed method are demonstrated by a continuous stirred tank heater case and a practical industrial system.**

***Note to Practitioners*—Multimode process monitoring is increasingly significant as industrial systems generally operate in varying operating conditions. However, most researches focus on multiple local monitoring models for complex multimode processes and assume that data of all possible modes are available and stored before learning. When similar or new modes arrive, local models are rebuilt corresponding to each mode and the model's capacity would increase with the continuous emergence of modes. Adaptive methods are a branch of multimode monitoring algorithms, but they strive to extract information of the current mode to ensure the monitoring performance while forgetting the previously learned knowledge gradually. This paper proposes a novel sparse dynamic inner principal component analysis with continual learning ability for multimode dynamic process monitoring, where modified synaptic intelligence is developed to measure the parameter importance accurately. It requires limited computation and storage resources for successive modes, which is convenient for practical applications. Similar to current multimode process monitoring algorithms, a set of data should be collected before learning a new mode, which may bring difficulties to real–time monitoring. For industrial systems, such as large-scale power plants and chemical systems, the proposed method has outstanding ability to monitor successive dynamic modes.**

***Index Terms*—Multimode dynamic process monitoring, sparse dynamic inner principal component analysis, modified synaptic intelligence, continual learning ability**

This work was supported by National Natural Science Foundation of China [grant numbers 62033008, 61873143]. (Corresponding authors: Donghua Zhou; Maoyin Chen)

Jingxin Zhang and Maoyin Chen are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zjx18@mails.tsinghua.edu.cn, mychen@tsinghua.edu.cn).

Donghua Zhou is with College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266000, China and also with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zdh@mail.tsinghua.edu.cn).

Xia Hong is with Department of Computer Science, School of Mathematical, Physical and Computational Sciences, University of Reading, RG6 6AY, U.K.

I. Introduction

For the sake of enhanced operational safety, advanced sensor technologies and data-driven process monitoring have received widespread attention in complex industrial systems [1]. These systems are dynamic in nature and the internal variables are time-correlated [2]. There are many researches devoted to dynamic process monitoring [3]–[5]. Canonical variate analysis (CVA) was proposed for nonlinear dynamic processes, where a state space model was established [3]. Two layered mixture Bayesian probabilistic principal component analysis (PCA) was presented to deal with non-Gaussian dynamic data, where data were divided into several clusters and a mixture model was built within each cluster [6]. Besides, dynamic inner PCA (DiPCA) was introduced to extract dynamic and static latent variables and adopted for dynamic process monitoring [7], [8].

Industrial systems generally operate under multiple modes due to changing of raw materials, market demands, etc [9]–[11]. Therefore, multimode process monitoring has undergone tremendous development recently [12]–[14], which can be divided into single–model schemes and multiple–model approaches [10], [15]. Most single–model methods remove the multimodality features by a transformation function [11], [16]–[18] and then establish a single monitoring model. When a novel mode arrives, the transformation function needs to be relearned. Alternatively, adaptive models are built to track the multiple data distribution based on the successive data [19]. However, they fail to monitor the varying modes because the features may change dramatically in the entire dataset and adaptive methods are effective for slow changing data [20]. Adaptive methods forget the previous information gradually to leave more space for the current mode. When the previously similar modes appear, the relevant learned knowledge has been overlapped and the model is difficult to track the changes quickly [11], [18]. Furthermore, enough data are required to update the model adaptively when a novel or similar mode arrives [21], [22]. Multiple–model schemes have been widely researched and in general these simply extend the aforementioned (single mode) dynamic monitoring algorithms

where a local model is built within each corresponding mode [14], [23]–[25]. For instance, structure dictionary learning was investigated for process monitoring and mode identification, where the common pattern and mode–specific pattern of each mode were extracted [14]. An improved mixture of probabilistic PCA (IMPPCA) could be utilized for multimode monitoring [26], where the mode could be identified by the posterior probability automatically. In [12], the multiple modes were separated by lazy learning and the residuals between the actual and predicted outputs were employed to detect faults for nonlinear dynamic processes. Mixture of CVA (MCVA) was developed for multimode dynamic processes, where the augmented data were divided by Gaussian mixture models (GMMs) and the local CVA model was built within each GMM [27]. Besides, dynamic latent variable model was proposed for multimode processes, where hidden Markov model was adopted to identify mode [2]. However, the aforementioned multiple–model methods require that historical data cover all possible modes, which is expensive and time-consuming [10], [14]. In practical applications, novel modes may appear especially for the systems with a short operating time. Moreover, similar modes may also occur continually. When a new or similar mode arrives, sufficient samples should be collected and a local model is established to monitor the corresponding mode. In the worst case, the historical data are required to be stored before learning and the monitoring model may be retrained from scratch [14], [26]. Thus, the model's capacity must increase significantly to cope with the successive emergence of modes, and so do the associated computational and storage costs. Therefore, it is essential to investigate effective methods for monitoring successive similar or novel modes, where the computation time and storage resources are constrained while acceptable monitoring performance is guaranteed.

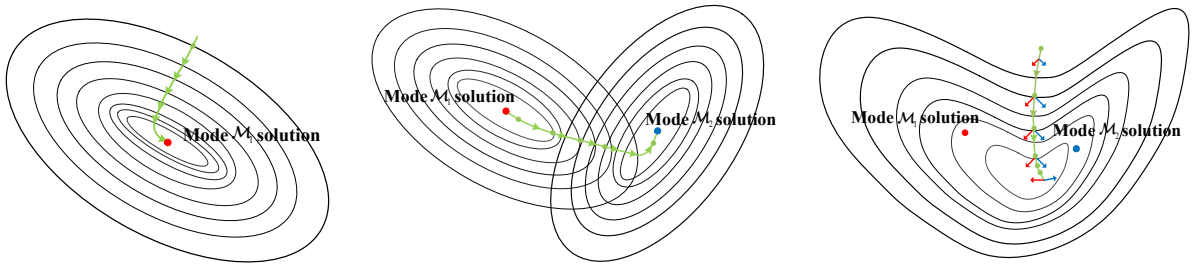
Similar to learning successive modes, humans and animals learn new knowledge continually from novel sequential tasks [28] and the previously learned information is still preserved simultaneously. Besides, the previous knowledge is beneficial for learning the new information quickly when the tasks are similar to some extent. Motivated by this learning manner, continual learning has received wide attention recently [29]–[33], which could potentially lend to solutions in multimode process monitoring algorithms. Specifically for successive modes, where the new modes are allowed to emerge continuously and the model could adapt to the changing modes [29]–[31]. Continual learning aims to learn the model continually by assimilating new information while preserving the learned knowledge [28]. The fundamental issue addressed by continual learning is catastrophic forgetting, where the previously learned knowledge may be interfered by new information and the performance is reduced on previous modes [28], [32], [33]. The concept of intelligent synapses in continual learning is aimed at adapting new information while storing the previously learned features. The superiorities of continual learning were analyzed in [15] and then continual learning was applied it to multimode process monitoring, where elastic weight consolidation (EWC) settled the catastrophic forgetting issue of PCA for successive modes and the prominent performance was acquired. The proposed method is short for PCA–EWC. However, it assumes that data follow multivariate Gaussian distribution and are stationary in each mode. The importance measure depends on the point estimate of Fisher information matrix [15], [30], which is naturally intractable in most cases.

Against this background, this paper has proposed a novel algorithm, referred to as sparse dynamic inner principal component analysis with modified synaptic intelligence (SDiPCA–MSI). To our best knowledge, DiPCA furnishes interpretability and prediction performance [7], [8], but relevant algorithms for multimode process monitoring are still deficient. Sparse representation is relevant for enhancing the model interpretability and reduce forgetting, where mode–sensitive parameters or effective changes to parameters are fewer [29], [33]. Therefore, we consider sparse DiPCA (SDiPCA) with continual learning ability for multimode dynamic monitoring, where the model captures new information and retains the learned knowledge simultaneously. Intuitively, the proposed method consolidates the changes in important parameters to monitor past modes, while allowing the unimportant parameters to learn the future modes. Moreover, modified synaptic intelligence (MSI) is proposed to evaluate the parameter importance, which is calculated along the learning trajectory and equivalent to the sensitivity to the loss [34], [35]. Similar to traditional multimode process monitoring approaches [21]–[23], [26], [27] and PCA–EWC [15], when a new mode arrives, a set of data onto the new mode should be collected before updating the SDiPCA–MSI model, thus delivering difficulty to unseen modes for online implementation. Besides, the significant information of existing modes is preserved when a new mode is trained, which allows it monitor the previous modes accurately based on one model and may also aid the learning of future relevant modes.

The contributions of this paper are summarized as follows:

a) Based on SDiPCA, this work investigates the continual learning ability for multimode dynamic processes, where the number of modes and samples per mode cannot be known a priori. The model is updated continually by consolidating new information while preserving the previously learned knowledge, thus the computation and storage expenses are limited with the increasing number of modes.
b) This paper proposes a novel estimation of importance measure, which is obtained by the intermediate parameters during the learning process. Compared with traditional synaptic intelligence (SI), MSI is more accurate and the random initialization makes it more feasible to acquire the optimal SDiPCA–MSI model parameters.

The rest of this paper is organized as follows. Section II initially presents the framework of DiPCA and basic concept of SI–based continual learning, followed by a formal problem statement of this paper. Section III presents the procedure of the proposed method for successive modes. Comparative experiments, computational complexity, strengths and potential limitations are discussed in Section IV. A continuous stirred tank heater (CSTH) process and a practical coal pulverizing system are adopted to illustrate the effectiveness in Section V. The concluding remarks are given in Section VI.



(a) (b) (c)

Fig. 1. Illustration of gradient decent optimization: (a) The trajectory for mode $\mathcal{M}_1$; (b) The trajectory when training the same model on the second mode $\mathcal{M}_2$ subsequently; (c) The trajectory when minimizing the summed loss from both modes (green) and gradients from each mode (red and blue) [29].

II. Problem Statement

A. DiPCA preliminaries

DiPCA [7] is aimed at extracting a set of dynamic latent variables that explain most dynamic variations of the data. Let $\boldsymbol{x} \in R^m$ be a sample vector of m variables and data matrix $\boldsymbol{X} \in R^{N\times m}$ with N samples. DiPCA starts with extracting the most predictable information by a vector autoregressive (VAR) model, referred to as the *inner* model, so that the dynamic relationship is characterized. Define the latent variables as

$$t_k = \boldsymbol{x}_k \boldsymbol{w} \tag{1}$$

where $\boldsymbol{w} \in R^m$ is the weight vector with $\|\boldsymbol{w}\|_2 = 1$. The current latent variables could be described by the past ones, namely,

$$t_k = \sum_{i=1}^{s} \beta_i t_{k-i} + r_k \tag{2}$$

where r_k is the Gaussian white noise at kth instant, s is the autoregressive order, or the number of time lags. Let $\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_s]^T$, $\|\boldsymbol{\beta}\|_2 = 1$. Based on (1) and (2), the prediction of the dynamic inner model is presented by:

$$\begin{aligned}\hat{t}_k &= \sum_{i=1}^{s} \boldsymbol{x}_{k-i}\boldsymbol{w}\beta_i \\ &= [\boldsymbol{x}_{k-1} \ \cdots \ \boldsymbol{x}_{k-s}]\,(\boldsymbol{\beta}\otimes\boldsymbol{w})\end{aligned}$$

where $\otimes$ denotes the Kronecker product. The dynamic latent variables are extracted by maximizing the covariance between t_k and $\hat{t}_k$, namely,

$$\frac{1}{N-s}\sum_{k=s+1}^{N} \boldsymbol{w}^T \boldsymbol{x}_k^T [\boldsymbol{x}_{k-1} \ \cdots \ \boldsymbol{x}_{k-s}]\,(\boldsymbol{\beta}\otimes\boldsymbol{w}) \tag{3}$$

Construct the matrices from $\boldsymbol{X}$,

$$\boldsymbol{X}^{(i)} = \left[\boldsymbol{x}_i^T \ \ \boldsymbol{x}_{i+1}^T \ \cdots \ \boldsymbol{x}_{N-s+i-1}^T\right]^T, \quad i = 1,\ldots,s+1 \tag{4}$$

$$\boldsymbol{Z} = \frac{1}{N-s}\left[\boldsymbol{X}^{(1)} \ \ \boldsymbol{X}^{(2)} \ \cdots \ \boldsymbol{X}^{(s)}\right] \tag{5}$$

Let $\boldsymbol{X}^{(s+1)}$ be denoted as $\tilde{\boldsymbol{X}}$, the objective of maximizing (3) is equivalent to

$$\begin{aligned}\min \quad & J(\boldsymbol{w},\boldsymbol{\beta}) = -\boldsymbol{w}^T\tilde{\boldsymbol{X}}^T\boldsymbol{Z}\,(\boldsymbol{\beta}\otimes\boldsymbol{w}) \\ s.t. \quad & \|\boldsymbol{w}\|_2 = 1, \ \ \|\boldsymbol{\beta}\|_2 = 1\end{aligned} \tag{6}$$

A recursively reduced data set is defined based on $\boldsymbol{X}$ and the optimization issue (6) is repeated until extracting l dynamic latent variables. In this paper, s and l are determined by [7].

B. Sparse DiPCA with SI for multimode dynamic processes

This work investigates SDiPCA with continual learning ability for multimode dynamic processes, without access to entire data sets. Moreover, the proposed algorithm aims to simultaneously monitor all modes with improved interpretation.

Denote multiple modes $\mathcal{M}_K$, $K = 1, 2, \ldots$. For ease of exposition, in each mode $\mathcal{M}_K$, it is assumed there are N_K samples in data set matrix $\boldsymbol{X}_K \in R^{N_K\times m}$ and the dynamic order s is the same for all modes. Clearly, with $\boldsymbol{X}$ set as $\boldsymbol{X}_K$, from which $\tilde{\boldsymbol{X}}_K$ and $\boldsymbol{Z}_K$ can be constructed by (4) and (5). Define the local SDiPCA cost function for the Kth mode, as

$$\tilde{J}_K(\boldsymbol{\theta},\boldsymbol{X}_K) = -\boldsymbol{w}^T\tilde{\boldsymbol{X}}_K^T\boldsymbol{Z}_K\,(\boldsymbol{\beta}\otimes\boldsymbol{w}) + \lambda_1\|\boldsymbol{w}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_1 \tag{7}$$

in which $\boldsymbol{\theta} = [\boldsymbol{w}^T, \boldsymbol{\beta}^T]^T$, λ_1 and λ_2 are associated positive regularization parameters. Increasing λ_1 and λ_2 will drive some parameters in $\boldsymbol{w}$ or $\boldsymbol{\beta}$ to zeros, leading to a sparse model.

The objective function (7) is directly based on data from each single mode, which means that at any time only a single mode is covered by the model parameters. To build a model that is able to track all modes, a novel composite cost function $J(\boldsymbol{\theta})$ will be defined by employing continual learning [33]–[35] and a novel efficient algorithm is proposed in Section III for multimode SDiPCA, which can adapt to a new mode without forgetting information learned from old modes, under the constraint that data from all past modes are not accessible.

For convenience, it is assumed that $J(\boldsymbol{\theta})$ has been defined as appropriate, we now briefly introduce the SI algorithm of combating forgetting in terms of parameter optimization trajectory in changing data environment. The gradient–based methods are efficient to acquire the optimal parameter $\boldsymbol{\theta}$. SI calculates the importance measure for each parameter along the optimization trajectory, which reflects the sensitivity of each parameter to the loss [34]. Given an infinitesimal update $\boldsymbol{\delta}(k)$ at kth iteration, the change in loss is approximated by

$$J(\boldsymbol{\theta}(k) + \boldsymbol{\delta}(k)) - J(\boldsymbol{\theta}(k)) \approx \sum\nolimits_i g_i(k)\,\delta_i(k)$$

where $\boldsymbol{g} = \frac{\partial J}{\partial \boldsymbol{\theta}}$ is the gradient, $\delta_i(k) = \theta_i(k) - \theta_i(k-1)$. The change in loss over the entire trajectory is calculated by

Algorithm 1 Update parameter by APG: $[\Theta_{k+1}] = F\left(\Theta_k, \boldsymbol{\varphi}_{k-1}, t_k, t_{k-1}, \alpha^{\varphi}, \nabla_{\boldsymbol{\varphi}} g\left(\cdot\right), g\left(\cdot\right)\right)$

1: $\boldsymbol{y}_k = \boldsymbol{\varphi}_k + \frac{t_{k-1}}{t_k}\left(\boldsymbol{z}_k - \boldsymbol{\varphi}_k\right) + \frac{t_{k-1}-1}{t_k}\left(\boldsymbol{\varphi}_k - \boldsymbol{\varphi}_{k-1}\right)$
2: Calculate $\boldsymbol{z}_{k+1} = prox_{h,t^y}\left(\boldsymbol{y}_k - t_k^y \nabla_{\boldsymbol{\varphi}} g\left(\boldsymbol{y}_k\right)\right)$ by (38) and (39), $t_k^y = f\left(\alpha^{\varphi}, v_k^y, \nabla_{\boldsymbol{\varphi}} g\left(\boldsymbol{y}_k\right)\right)$ by (21), and v_k^y is updated by (22);
3: Calculate $\boldsymbol{v}_{k+1} = prox_{h,t^{\varphi}}\left(\boldsymbol{\varphi}_k - t_k^{\varphi} \nabla_{\boldsymbol{\varphi}} g\left(\boldsymbol{\varphi}_k\right)\right)$ by (38) and (39), $t_k^{\varphi} = f\left(\alpha^{\varphi}, v_k^{\varphi}, \nabla_{\boldsymbol{\varphi}} g\left(\boldsymbol{\varphi}_k\right)\right)$ by (21), and v_k^{φ} is updated by (22);
4: $\boldsymbol{\varphi}_{k+1} = \begin{cases} \boldsymbol{z}_{k+1}, & if\ g\left(\boldsymbol{z}_{k+1}\right) \le g\left(\boldsymbol{v}_{k+1}\right) \\ \boldsymbol{v}_{k+1}, & otherwise \end{cases}$

$$\begin{aligned}\sum_k \boldsymbol{g}\left(k\right)^T \boldsymbol{\delta}\left(k\right) =& \sum_i \sum_k g_i\left(k\right) \delta_i\left(k\right) \\ =& -\sum_i \varpi_i\end{aligned}$$

Equivalently,

$$\varpi_i = \sum_k \left(\theta_i\left(k\right) - \theta_i\left(k-1\right)\right) \frac{-\partial J}{\partial \theta_i\left(k\right)} \tag{8}$$

In multimode dynamic process monitoring using SI, consider the data block is received in the order of their mode index. When Kth mode arrives ($K \ge 2$), data of current mode and the previously learned model parameters are available. It is intractable to acquire the true summed loss of previous modes based on all data. In this paper, a surrogate loss J_{total}^{K-1} is introduced to approximate the total loss of previous $(K-1)$ modes [34]. During the stage of Kth mode, our loss function $J(\boldsymbol{\theta})$ of K modes under SI framework is approximated by

$$\begin{aligned}J_{total}^{K}\left(\boldsymbol{\theta}\right) =& \tilde{J}_K\left(\boldsymbol{\theta}, \boldsymbol{X}_K\right) + \gamma_K J_{total}^{K-1}\left(\hat{\boldsymbol{\theta}}^{K-1}, \hat{\boldsymbol{\varpi}}_{K-1}\right) \\ =& \tilde{J}_K\left(\boldsymbol{\theta}, \boldsymbol{X}_K\right) + \gamma_K \sum_i \hat{\varpi}_i^{K-1}\left(\theta_i - \hat{\theta}_i^{K-1}\right)^2\end{aligned} \tag{9}$$

where $\hat{\boldsymbol{\theta}}^{K-1}$ and $\hat{\boldsymbol{\varpi}}_{K-1}$ are the estimated parameter vector and corresponding importance measure vector for the previous $K-1$ modes, with $\hat{\theta}_i^{K-1}$ and $\hat{\varpi}_i^{K-1}$ being the entries at the ith dimension, respectively. $\gamma_K > 0$ is the hyper-parameter that balances the previous modes and the current one. The choice of γ_K is data dependent. A larger γ_K maintains more information to previous modes. A smaller one is more adaptive to current mode, yielding to better local fit to current mode with better online performance.

The total objective (9) combines local recent mode objective $\tilde{J}_K$, with model of the past modes via model parameters $\hat{\theta}_i^{K-1}$ and $\hat{\varpi}_i^{K-1}$ represents the importance measure of θ_i for all previous modes, without access to past data. After learning the Kth mode, it is updated by (35) in Appendix A, namely,

$$\hat{\varpi}_i^K \approx \frac{\varpi_i^K}{\left(\hat{\theta}_i^K - \hat{\theta}_i^{K-1}\right)^2} + \hat{\varpi}_i^{K-1} \tag{10}$$

which is initialized as $\hat{\varpi}_i^0 = 0$. ϖ_i^K denotes the difference of losses before and after training θ_i for the Kth mode from (8), which accumulates the change of loss $\tilde{J}_K$ along the parameter optimization trajectory.

The illustration of applying SI to multimode dynamic processes is interpreted by considering two successive modes. The optimization trajectory of mode $\mathcal{M}_1$ is described in Fig. 1(a). When mode $\mathcal{M}_2$ arrives, the traditional manner only considers mode $\mathcal{M}_2$ if one model is built, thus leading to great loss on mode $\mathcal{M}_1$ in Fig. 1(b). Significantly, this work aims to build a continually updated model with continual learning ability, which preserves the learned knowledge and extracts the critical features of the current mode simultaneously. As depicted in Fig. 1(c), the total loss of two modes is considered to acquire the appropriate parameters for both modes. Without loss of generality, it can be extended to more successive modes.

III. SDiPCA–MSI for process monitoring

In this section, MSI is proposed firstly and then SDiPCA–MSI is presented. The algorithm deals with multimode data sets in consecutive order of modes $K = 1, 2, ...$ Hence, the model is developed initially when $K = 1$, followed by that of any mode ($K \ge 2$), which recursively updates the model from $(K-1)$ modes. Thus, the SDiPCA–MSI framework can be unified for any mode ($K \ge 1$). Finally, the training and associated online monitoring phases are summarized.

A. Modified SI

Clearly a key issue is to optimize (9) and estimate $\hat{\boldsymbol{\varpi}}$. Up to the Kth mode, the surrogate loss is defined as

$$J_{total}^{K}\left(\theta_i\right) = \hat{\varpi}_i^K\left(\theta_i - \hat{\theta}_i^K\right)^2$$

According to (36) in Appendix A, $\hat{\varpi}_i^K$ is calculated by

$$\begin{aligned}\hat{\varpi}_i^K =& \frac{J_{total}^{K}\left(\theta_i^0\right) - J_{total}^{K}\left(\hat{\theta}_i^K\right)}{\left(\theta_i^0 - \hat{\theta}_i^K\right)^2} \\ =& \frac{\varpi_i^K}{\left(\theta_i^0 - \hat{\theta}_i^K\right)^2}\end{aligned} \tag{11}$$

where ϖ_i^K is the difference of the losses J_{total}^K before and after training and can be computed by (8). The initial value θ_i^0 of the optimization problem can be selected randomly under the constraints. When $\theta_i^0 = \hat{\theta}_i^{K-1}$, the importance measure is calculated by (37).

It has been illustrated that (11) is more accurate than (10) in Appendix A. Compared with SI [34], MSI provides two distinctions: a) MSI utilizes the total loss (9) of all modes to estimate the importance measure while SI uses the loss (7) of the current mode. Because (9) is the exact optimization objective, it is easier to calculate than (7); b) The initialization of MSI is random while the initial setting of SI is the optimal value of the last mode. Since the objective of SDiPCA–MSI is nonconvex and nonconcvae, random initialization is beneficial for seeking the appropriate model parameters.

Algorithm 2 Optimization procedure of SDiPCA–MSI

Input: Data $\boldsymbol{X}_K$, l, s, paramaters of mode $\mathcal{M}_{K-1}$ $\left\{\boldsymbol{W}_{\mathcal{M}_{K-1}}, \boldsymbol{\Gamma}_{\mathcal{M}_{K-1}}, \boldsymbol{\Omega}_{\mathcal{M}_{K-1}}\right\}$

Output: Weight matrix $\boldsymbol{W}_{\mathcal{M}_K}$, regression coefficient matrix $\boldsymbol{\Gamma}_{\mathcal{M}_K}$, the importance measure $\boldsymbol{\Omega}_{\mathcal{M}_K}$, projection matrix $\boldsymbol{P}_{\mathcal{M}_K}$

Initialization: Random unit vectors $\boldsymbol{\beta}_0$ and $\boldsymbol{w}_0$

for $j = 1, 2, \cdots, l$ **do**

1) Let $\boldsymbol{w}^{K-1}$, $\boldsymbol{\beta}^{K-1}$, $\hat{\boldsymbol{\varpi}}$ be the jth line of $\boldsymbol{W}_{\mathcal{M}_{K-1}}$, $\boldsymbol{\Gamma}_{\mathcal{M}_{K-1}}$ and $\boldsymbol{\Omega}_{\mathcal{M}_{K-1}}$, respectively. $\bar{\boldsymbol{\Omega}} = diag(\hat{\boldsymbol{\varpi}})$, $\hat{\boldsymbol{\theta}}^{K-1} = \left[\left(\boldsymbol{w}^{K-1}\right)^T, \left(\boldsymbol{\beta}^{K-1}\right)^T\right]^T$;
2) Initialize $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0$, $\boldsymbol{w}_1 = \boldsymbol{w}_0$, $\boldsymbol{z}_1^w = \boldsymbol{w}_0$, $\boldsymbol{z}_1^\beta = \boldsymbol{\beta}_0$, $\boldsymbol{\varpi}_1 = \boldsymbol{0}$, $t_1 = 1$, $t_0 = 0$, $\alpha^w = 0.005$, $\alpha^\beta = 0.005$, $\alpha^\nu = 0.2$, $t_0^w = t_0^{wy} = 10^{-4}$, $t_0^\beta = t_0^{\beta y} = 10^{-4}$, $\nu_{1,1} = 0$, $\nu_{2,1} = 0$, $v_0^{wy} = 0$, $v_0^w = 0$, $v_0^{\beta y} = 0$, $v_0^\beta = 0$, $k = 1$;
3) Let $\Theta_1^w = \left\{\boldsymbol{w}_1, \boldsymbol{z}_1^w, t_0^{wy}, t_0^w, v_0^{wy}, v_0^w\right\}$ and $\Theta_1^\beta = \left\{\boldsymbol{\beta}_1, \boldsymbol{z}_1^\beta, t_0^{\beta y}, t_0^\beta, v_0^{\beta y}, v_0^\beta\right\}$, construct the objective function according to (13) and (14);
4) Calculate optimal $\boldsymbol{w}$ and $\boldsymbol{\beta}$ based on the predicted covariation:

while the issue (13) is not converged **do**

a) Update $\boldsymbol{w}$, $[\Theta_{k+1}^w] = F\left(\Theta_k^w, \boldsymbol{w}_{k-1}, t_k, t_{k-1}, \alpha^w, \nabla_{\boldsymbol{w}} g\left(\boldsymbol{w}_k, \boldsymbol{\beta}_k\right), g(\boldsymbol{w}_k, \boldsymbol{\beta}_k)\right)$ in Algorithm 1, $g(\boldsymbol{w}_k, \boldsymbol{\beta}_k)$ and $\nabla_{\boldsymbol{w}} g\left(\boldsymbol{w}_k, \boldsymbol{\beta}_k\right)$ are calculated by (15) and (17) respectively;
b) Update $\boldsymbol{\beta}$, $[\Theta_{k+1}^\beta] = F\left(\Theta_k^\beta, \boldsymbol{\beta}_{k-1}, t_k, t_{k-1}, \alpha^\beta, \nabla_{\boldsymbol{\beta}} g\left(\boldsymbol{w}_{k+1}, \boldsymbol{\beta}_k\right), g(\boldsymbol{w}_{k+1}, \boldsymbol{\beta}_k)\right)$ in Algorithm 1, $g(\boldsymbol{w}_{k+1}, \boldsymbol{\beta}_k)$ and $\nabla_{\boldsymbol{\beta}} g\left(\boldsymbol{w}_{k+1}, \boldsymbol{\beta}_k\right)$ are calculated by (15) and (18) respectively;
c) $t_{k+1} = \frac{\sqrt{4(t_k)^2+1}+1}{2}$;
d) Update Lagrange parameters: $\nu_{1,k+1} = \nu_{1,k} + \alpha^\nu \nabla_{\nu_1} g\left(\boldsymbol{w}_{k+1}\right)$, $\nu_{2,k+1} = \nu_{2,k} + \alpha^\nu \nabla_{\nu_2} g\left(\boldsymbol{\beta}_{k+1}\right)$;
e) Construct $\boldsymbol{\theta}_{k+1} = [\boldsymbol{w}_{k+1}^T, \boldsymbol{\beta}_{k+1}^T]^T$, $\nabla_{\boldsymbol{\theta}} g\left(\boldsymbol{\theta}_{k+1}\right) = \left[\left(\nabla_{\boldsymbol{w}} g(\boldsymbol{w}_{k+1}, \boldsymbol{\beta}_{k+1})\right)^T, \left(\nabla_{\boldsymbol{\beta}} g(\boldsymbol{w}_{k+1}, \boldsymbol{\beta}_{k+1})\right)^T\right]^T$;
f) Calculate the importance measure: $\boldsymbol{\varpi}_{k+1} = \boldsymbol{\varpi}_k - \left(\left(\nabla_{\boldsymbol{\theta}} g\left(\boldsymbol{\theta}_{k+1}\right)\right)^T \odot \left(\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\right)^T\right)^T$;
g) $k = k + 1$;

end while

5) Let $\boldsymbol{t} = \boldsymbol{X}_K \boldsymbol{w}$, $\boldsymbol{p} = \boldsymbol{X}_K^T \boldsymbol{t} / \boldsymbol{t}^T \boldsymbol{t}$. Deflate $\boldsymbol{X}_K$ as $\boldsymbol{X}_K := \boldsymbol{X}_K - \boldsymbol{t}\boldsymbol{p}^T$;
6) The importance measure is normalized by (24) and denoted as $\hat{\boldsymbol{\varpi}}_j$. The loading vector is labeled as $\boldsymbol{p}_j$, the weighted vector is $\boldsymbol{w}_j^*$, the regression coefficient is $\boldsymbol{\beta}_j^*$. Let $\boldsymbol{w}_0 = \boldsymbol{w}_j^*$, $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_j^*$;

end for

$\boldsymbol{W}_{\mathcal{M}_K} = [\boldsymbol{w}_1^* \ \boldsymbol{w}_2^* \ \cdots \ \boldsymbol{w}_l^*]$, $\boldsymbol{\Gamma}_{\mathcal{M}_K} = [\boldsymbol{\beta}_1^* \ \boldsymbol{\beta}_2^* \ \cdots \ \boldsymbol{\beta}_l^*]$, $\boldsymbol{\Omega}_{\mathcal{M}_K} = [\hat{\boldsymbol{\varpi}}_1 \ \hat{\boldsymbol{\varpi}}_2 \ \cdots \ \hat{\boldsymbol{\varpi}}_l]$, $\boldsymbol{P}_{\mathcal{M}_K} = [\boldsymbol{p}_1 \ \boldsymbol{p}_2 \ \cdots \ \boldsymbol{p}_l]$

For convenience, we use the vector form to describe the importance measure, namely,

$$\boldsymbol{\varpi} = \sum_k \left(\left(-\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}_k\right)\right)^T \odot \left(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}\right)^T\right)^T \tag{12}$$

where $\odot$ denotes the Khatri-Rao product.

B. SDiPCA–MSI

When the mode $\mathcal{M}_K$ appears ($K \geq 1$), data $\boldsymbol{X}_K$ are collected. According to (9), the objective is

$$\begin{aligned} \min_{\boldsymbol{\theta}} \ & J_{total}^K(\boldsymbol{\theta}) = \tilde{J}_K\left(\boldsymbol{\theta}, \boldsymbol{X}_K\right) + \gamma_K J_{total}^{K-1}\left(\hat{\boldsymbol{\theta}}^{K-1}, \hat{\boldsymbol{\varpi}}^{K-1}\right) \\ s.t. \ & \boldsymbol{w}^T \boldsymbol{w} = 1, \boldsymbol{\beta}^T \boldsymbol{\beta} = 1 \end{aligned} \tag{13}$$

where $\tilde{J}_K\left(\boldsymbol{\theta}, \boldsymbol{X}_K\right)$ is calculated by (7) and

$$\begin{aligned} & \gamma_K J_{total}^{K-1}\left(\hat{\boldsymbol{\theta}}^{K-1}, \hat{\boldsymbol{\varpi}}^{K-1}\right) \\ =& \gamma_{1,K}\left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right)^T \bar{\boldsymbol{\Omega}}_{\boldsymbol{w}} \left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right) \\ & + \gamma_{2,K}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{K-1}\right)^T \bar{\boldsymbol{\Omega}}_{\boldsymbol{\beta}} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{K-1}\right) \end{aligned}$$

in which $\hat{\boldsymbol{\theta}}^{K-1} = \left[\left(\boldsymbol{w}^{K-1}\right)^T, \left(\boldsymbol{\beta}^{K-1}\right)^T\right]^T$ is the optimal parameter of mode $\mathcal{M}_{K-1}$. $\hat{\boldsymbol{\varpi}}_{\boldsymbol{w}}^{K-1} \in R^m$ and $\hat{\boldsymbol{\varpi}}_{\boldsymbol{\beta}}^{K-1} \in R^s$ are the importance measures of $\boldsymbol{w}$ and $\boldsymbol{\beta}$ respectively. Then, $\hat{\boldsymbol{\varpi}}^{K-1} = \left[\left(\hat{\boldsymbol{\varpi}}_{\boldsymbol{w}}^{K-1}\right)^T, \left(\hat{\boldsymbol{\varpi}}_{\boldsymbol{\beta}}^{K-1}\right)^T\right]^T$, $\bar{\boldsymbol{\Omega}}_{\boldsymbol{w}} = diag(\hat{\boldsymbol{\varpi}}_{\boldsymbol{w}}^{K-1})$, $\bar{\boldsymbol{\Omega}}_{\boldsymbol{\beta}} = diag(\hat{\boldsymbol{\varpi}}_{\boldsymbol{\beta}}^{K-1})$ and $\bar{\boldsymbol{\Omega}} = diag(\hat{\boldsymbol{\varpi}}^{K-1})$. $\gamma_{1,K}$ and $\gamma_{2,K}$ are the regularization coefficients about $\boldsymbol{w}$ and $\boldsymbol{\beta}$.

The augmented Lagrange function of (13) is depicted as

$$\begin{aligned} & \tilde{J}_{total}^K(\boldsymbol{\theta}, \boldsymbol{X}_K) \\ =& -\boldsymbol{w}^T \tilde{\boldsymbol{X}}_K^T \boldsymbol{Z}_K\left(\boldsymbol{\beta} \otimes \boldsymbol{w}\right) + \nu_1\left(\boldsymbol{w}^T \boldsymbol{w} - 1\right)^2 + \nu_2\left(\boldsymbol{\beta}^T \boldsymbol{\beta} - 1\right)^2 \\ & + \lambda_1 \|\boldsymbol{w}\|_1 + \gamma_{1,K}\left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right)^T \bar{\boldsymbol{\Omega}}_{\boldsymbol{w}} \left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right) \\ & + \lambda_2 \|\boldsymbol{\beta}\|_1 + \gamma_{2,K}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{K-1}\right)^T \bar{\boldsymbol{\Omega}}_{\boldsymbol{\beta}} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{K-1}\right) \end{aligned} \tag{14}$$

The issue (14) is nonconvex and nonsmooth, which is intractable to compute the gradient directly. Then, we adopt accelerated proximal gradient (APG) to optimize (14) [36].

APG algorithm: The proximal gradient descent algorithm is shown in Appendix B for reference. Similar to proximal gradient descent, (14) is divided into the smooth part $g_K(\boldsymbol{w}, \boldsymbol{\beta})$ and the nonsmooth part $h(\boldsymbol{w}, \boldsymbol{\beta})$.

$$\begin{aligned} & g_K(\boldsymbol{w}, \boldsymbol{\beta}) \\ =& -\boldsymbol{w}^T \tilde{\boldsymbol{X}}_K^T \boldsymbol{Z}_K\left(\boldsymbol{\beta} \otimes \boldsymbol{w}\right) + \nu_1\left(\boldsymbol{w}^T \boldsymbol{w} - 1\right)^2 \\ & + \gamma_{1,K}\left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right)^T \bar{\boldsymbol{\Omega}}_{\boldsymbol{w}} \left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right) + \nu_2\left(\boldsymbol{\beta}^T \boldsymbol{\beta} - 1\right)^2 \\ & + \gamma_{2,K}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{K-1}\right)^T \bar{\boldsymbol{\Omega}}_{\boldsymbol{\beta}} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{K-1}\right) \end{aligned} \tag{15}$$

$$h(\boldsymbol{w}, \boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_1 \tag{16}$$

Accordingly, the gradients with respect to parameters are

$$\begin{aligned} \nabla_{\boldsymbol{w}} g = \frac{\partial g_K}{\partial \boldsymbol{w}} =& -\left(\boldsymbol{G}_{K,\boldsymbol{\beta}} + \boldsymbol{G}_{K,\boldsymbol{\beta}}^T\right)\boldsymbol{w} + 4\nu_1 \boldsymbol{w}\left(\boldsymbol{w}^T \boldsymbol{w} - 1\right) \\ & + 2\gamma_{1,K} \bar{\boldsymbol{\Omega}}_{\boldsymbol{w}}\left(\boldsymbol{w} - \boldsymbol{w}^{K-1}\right) \end{aligned} \tag{17}$$

TABLE I
DISCUSSION OF PARAMETERS

Parameters	Illustrations
$l=0$	SDiPCA is transformed to SPCA, and only static latent variables are considered.
$l=m$	Only dynamic latent variables are extracted.
$\lambda_1=0$, $\lambda_2=0$	SDiPCA is transformed to traditional DiPCA.
$\gamma_{1,K}=0$, $\gamma_{2,K}=0$	SDiPCA–MSI is transformed to SDiPCA.
$\gamma_{1,K}\to\infty$, $\gamma_{2,K}\to\infty$	Information of current mode $\mathcal{M}_K$ is neglected and the learned knowledge of previous modes is preserved.

$$\nabla_{\boldsymbol{\beta}} g = \frac{\partial g_K}{\partial \boldsymbol{\beta}} = -(\boldsymbol{I}_s \otimes \boldsymbol{w})^T \boldsymbol{Z}_K^T \tilde{\boldsymbol{X}}_K \boldsymbol{w} + 4\nu_2 \boldsymbol{\beta}\left(\boldsymbol{\beta}^T\boldsymbol{\beta}-1\right) + 2\gamma_{2,K}\bar{\boldsymbol{\Omega}}_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^{K-1}\right) \tag{18}$$

$$\nabla_{\nu_1} g = \frac{\partial g_K}{\partial \nu_1} = \left(\boldsymbol{w}^T\boldsymbol{w}-1\right)^2 \tag{19}$$

$$\nabla_{\nu_2} g = \frac{\partial g_K}{\partial \nu_2} = \left(\boldsymbol{\beta}^T\boldsymbol{\beta}-1\right)^2 \tag{20}$$

where $\boldsymbol{G}_{K,\boldsymbol{\beta}} = \tilde{\boldsymbol{X}}_K^T \boldsymbol{Z}_K (\boldsymbol{\beta}\otimes \boldsymbol{I}_m)$.

For the nonsmooth part h, the proximal gradient of the L_1 regularization term is determined by soft threshold [37] and calculated by (39). Motivated by Adam [38], the step size is calculated adaptively to accelerate the convergence rate, namely,

$$t_k = f(\alpha, \upsilon_k, \nabla g_k) = \alpha \Big/ \left(\sqrt{\left(\tau_2\upsilon_k + (1-\tau_1)\left\|\nabla g_k\right\|^2\right)\Big/(1-\tau_2)} + \varepsilon\right) \tag{21}$$

where υ_k is updated by

$$\upsilon_k = \tau_2\upsilon_{k-1} + (1-\tau_2)\|\nabla g_k\|^2 \tag{22}$$

in which ∇g_k is the corresponding gradient at kth iteration, α is constant, $\tau_1 = 0.9$ and $\tau_2 = 0.999$. ε is added to avoid ill–conditioning issues with $\varepsilon = 10^{-8}$. The procedure of APG is summarized in Algorithm 1.

***Modified synaptic intelligence**:* Recall (12), the importance measure is computed by

$$\boldsymbol{\varpi} = \sum_k \left(\left(-\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_k)\right)^T \odot \left(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}\right)^T\right)^T \tag{23}$$

where $\boldsymbol{\theta}_k = \left[\boldsymbol{w}_k^T, \boldsymbol{\beta}_k^T\right]^T$, $\boldsymbol{w}_k$ and $\boldsymbol{\beta}_k$ are the updated parameters after kth iteration. The objective function g is calculated by (15). $\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_k) = \left[\left(\nabla_{\boldsymbol{w}} g(\boldsymbol{w}_k, \boldsymbol{\beta}_k)\right)^T, \left(\nabla_{\boldsymbol{\beta}} g(\boldsymbol{w}_k, \boldsymbol{\beta}_k)\right)^T\right]^T$, which is calculated based on (17) and (18). After the training procedure, each element of $\boldsymbol{\varpi}$ is normalized by [33]

$$\hat{\varpi}_i = \max\left(0, \frac{\varpi_i}{\left(\Delta\theta_i\right)^2 + \zeta}\right) \tag{24}$$

where $\Delta\theta_i$ is the total change of ith variable for mode $\mathcal{M}_K$, $1 \le i \le s+m$. $\zeta > 0$ is added to avoid ill–conditioning issues. Similar to (11), the importance measure is normalized in (24) to ensure that the regularization term shares the same unit with the true objective function.

To make the dynamic latent variables mutually uncorrelated, the parameters of each latent variable are estimated greedily. The solution procedure for (13) is summarized as the inner–loop of Algorithm 2, where its out–loop iterates over the number of latent variables with index l. The optimal parameters are denoted as $\boldsymbol{W}_{\mathcal{M}_K}$, $\boldsymbol{\Gamma}_{\mathcal{M}_K}$, $\boldsymbol{P}_{\mathcal{M}_K}$, $\boldsymbol{\Omega}_{\mathcal{M}_K}$. Obviously, $\boldsymbol{\Theta}_{\mathcal{M}_K} = [\boldsymbol{W}_{\mathcal{M}_K}; \boldsymbol{\Gamma}_{\mathcal{M}_K}]$. After the training procedure finishes, we lose the access to data $\boldsymbol{X}_K$. Note that when $K=1$, $\boldsymbol{\Omega}_{\mathcal{M}_0} = \boldsymbol{0}$. There is no need to provide $\boldsymbol{W}_{\mathcal{M}_0}$ and $\boldsymbol{\Gamma}_{\mathcal{M}_0}$. Here, the total loss J_{total}^K is actually the loss function of mode $\mathcal{M}_1$ and Algorithm 2 is also applied to solve the optimization issue.

We discuss the influence of regularization coefficients and the number of dynamic latent variables l in Table I. Specifically, the hyperparameters $\gamma_{1,K}$ and $\gamma_{2,K}$ play an important role in distributing the importance of sequential modes. When the Kth mode is especially significant, the values of $\gamma_{1,K}$ and $\gamma_{2,K}$ would be small. Here we mainly focus on the performance of the current mode. If the current mode is regarded as unimportant by prior knowledge, the hyperparameters may be large to forget the current information gracefully.

C. Monitoring model

Define $\boldsymbol{R} = \boldsymbol{W}_{\mathcal{M}_K}\left(\boldsymbol{P}_{\mathcal{M}_K}^T\boldsymbol{W}_{\mathcal{M}_K}\right)^{-1}$, $\boldsymbol{T} = \boldsymbol{X}_K\boldsymbol{R}$. Form the $\boldsymbol{T}_i$, $i = 1, \cdots, s+1$, from $\boldsymbol{T}$ in the same way with (4). Similar to (2), we establish the relationship between the predictable latent scores $\boldsymbol{T}_{s+1}$ and the past $\boldsymbol{T}_1, \boldsymbol{T}_2, \cdots, \boldsymbol{T}_s$:

$$\boldsymbol{T}_{s+1} = \sum_{i=1}^{s} \boldsymbol{T}_i \boldsymbol{\Phi}_{s+1-i} + \boldsymbol{E} = \bar{\boldsymbol{T}}_s\boldsymbol{\Phi} + \boldsymbol{E} \tag{25}$$

where $\bar{\boldsymbol{T}}_s = [\boldsymbol{T}_1\ \boldsymbol{T}_2\ \cdots\ \boldsymbol{T}_s]$, $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_s\ \boldsymbol{\Phi}_{s-1}\ \cdots\ \boldsymbol{\Phi}_1]$. $\boldsymbol{\Phi}$ is estimated by original least squares, namely,

$$\hat{\boldsymbol{\Phi}} = \left(\bar{\boldsymbol{T}}_s^T\bar{\boldsymbol{T}}_s\right)^{-1}\bar{\boldsymbol{T}}_s^T\boldsymbol{T}_{s+1} \tag{26}$$

Algorithm 3 Training procedure of SDiPCA–MSI

1: Calculate the mean value and standard deviation of the training data, and normalize them with zero mean and unit variance;
2: For mode $\mathcal{M}_K$, solve the optimization issue (13) by Algorithm 2 and the parameters are updated by APG in Algorithm 1;
3: Build a VAR model for latent scores by (25)–(26) and $\boldsymbol{T}_{s+1}$ is predicted by (27);
4: Calculate the prediction error matrix $\boldsymbol{E}$ by (31) and perform PCA, namely, $\boldsymbol{E} = \boldsymbol{T}_r\boldsymbol{P}_r^T + \boldsymbol{E}_r$;
5: Calculate the monitoring statistics by (29) and (32), and the corresponding thresholds are estimated by KDE.

Algorithm 4 Online monitoring procedure of SDiPCA–MSI

1: Preprocess the testing data based on the mean value and standard deviation of normal training data;
2: Calculate the dynamic latent scores by (25), the predictable latent scores by (27) and the dynamic residual matrix by (28);
3: Calculate two monitoring statistics by (29) and (32);
4: Judge the operating condition: all statistics are lower than the thresholds, the process is normal. Otherwise, the process is faulty.

Then, $\boldsymbol{T}_{s+1}$ is predicted by

$$\hat{\boldsymbol{T}}_{s+1} = \bar{\boldsymbol{T}}_s \hat{\boldsymbol{\Phi}} \tag{27}$$

The dynamic residual matrix $\boldsymbol{V}$ is computed by:

$$\boldsymbol{V} = \boldsymbol{T} - \hat{\boldsymbol{T}}_{s+1} \tag{28}$$

Monitoring the latent dynamic score matrix $\boldsymbol{T}$ directly may lead to high false alarms. Thus, a monitoring index is built through $\boldsymbol{V}$ and defined as

$$T_{\varphi}^2 = \boldsymbol{v}^T \boldsymbol{\Phi}_v \boldsymbol{v} \tag{29}$$

$$\boldsymbol{\Phi}_v = \frac{\boldsymbol{P}_v \boldsymbol{\Lambda}_v^{-1} \boldsymbol{P}_v^T}{J_{th,T_v^2}} + \frac{\boldsymbol{I} - \boldsymbol{P}_v \boldsymbol{P}_v^T}{J_{th,\mathrm{SPE_v}}} \tag{30}$$

where $\boldsymbol{P}_v$ is the principal component matrix by PCA, and $\boldsymbol{\Lambda}_v = \frac{1}{N_K - s - 1} \boldsymbol{V}^T \boldsymbol{V}$. J_{th,T_v^2} and $J_{th,\mathrm{SPE_v}}$ are the thresholds of two statistics T_v^2 and $\mathrm{SPE_v}$ based on PCA, respectively.

The static prediction error is

$$\boldsymbol{E} = \boldsymbol{X}_{s+1} - \boldsymbol{T}_{s+1} \boldsymbol{P}^T \tag{31}$$

Similar to (29), an index is designed to monitor the static error

$$T_c^2 = \boldsymbol{e}^T \boldsymbol{\Phi}_c \boldsymbol{e} \tag{32}$$

$$\boldsymbol{\Phi}_c = \frac{\boldsymbol{P}_r \boldsymbol{\Lambda}_r^{-1} \boldsymbol{P}_r^T}{J_{th,T_r^2}} + \frac{\boldsymbol{I} - \boldsymbol{P}_r \boldsymbol{P}_r^T}{J_{th,\mathrm{SPE_r}}} \tag{33}$$

where $\boldsymbol{P}_r$ is the principal component matrix. Perform PCA on $\boldsymbol{E}$, then $\boldsymbol{E} = \boldsymbol{T}_r \boldsymbol{P}_r^T + \boldsymbol{E}_r$ and $\boldsymbol{\Lambda}_r = \frac{1}{N_K - s - 1} \boldsymbol{T}_r^T \boldsymbol{T}_r$. J_{th,T_r^2} and $J_{th,\mathrm{SPE_r}}$ are the thresholds of T_r^2 and $\mathrm{SPE_r}$.

The thresholds are calculated by kernel density estimation (KDE) [26]. When a novel mode arrives, similar to current multimode process monitoring methods, partial normal data should be collected and then the model is updated offline in Algorithm 3. Since the previously learned knowledge is relevant for learning the current mode, enough data are not required before learning [21], [22]. The monitoring phase is summarized in Algorithm 4.

IV. Discussion

A. Comparative experiments

Take three successive modes as an example to illustrate the superiorities of SDiPCA–MSI in Table II. Four typical multimode algorithms are selected for comparison, namely, PCA–EWC [15], RPCA [19], IMPPCA [26] and MCVA [27]. SDiPCA–MSI, PCA–EWC and RPCA are adaptive methods based on a single model, while IMPPCA and MCVA belong to multiple–model methods.

For Situations 1–11, we illustrate the catastrophic forgetting issue of SDiPCA and continual learning ability of SDiPCA–MSI for sequential modes. Specifically, we get the model A after training the mode $\mathcal{M}_1$ by SDiPCA. When the mode $\mathcal{M}_2$ arrives, a set of data should be collected and the model is updated by assimilating new information while retraining the learned features. Therefore, the model B is expected to monitor two modes simultaneously and then the backward transfer ability is reflected. For Situations 4 and 5, the comparison aims to interpret the catastrophic forgetting issue of SDiPCA, which reveals that the model for one mode underperforms for another mode. Hence, the performance of Situation 5 should be poor. In the most ideal case, the performance of Situation 2 is better than that of Situation 4, which indicates that the learned knowledge of previous modes may aid the learning of future relevant modes. Aforementioned analysis could also be applied to Situations 6–11. For PCA–EWC, the experimental procedures of Situations 12–17 are analogous to those of SDiPCA–MSI. We intend to compare the continual learning ability between PCA–EWC and SDiPCA–MSI. RPCA is utilized for comparison, as it fails to track the dramatic changes in multimode processes. The model is updated based on new data and the previous information is forgotten gradually.

Similar to most multiple–model algorithms, IMPPCA and MCVA identify the modes and build the local monitoring model within each mode. Data from all possible modes are required before learning, which is expensive and time-consuming. Take IMPPCA as an instance, MCVA shares

TABLE II
Comparative schemes for case study

	Methods	Training sources (Model + Data)	Model label	Testing sources
Situation 1	SDiPCA	$\mathcal{M}_1$	A	$\mathcal{M}_1$
Situation 2	SDiPCA–MSI	A + $\mathcal{M}_2$	B	$\mathcal{M}_2$
Situation 3	SDiPCA–MSI	-	B	$\mathcal{M}_1$
Situation 4	SDiPCA	$\mathcal{M}_2$	C	$\mathcal{M}_2$
Situation 5	SDiPCA	-	C	$\mathcal{M}_1$
Situation 6	SDiPCA–MSI	B + $\mathcal{M}_3$	D	$\mathcal{M}_3$
Situation 7	SDiPCA–MSI	-	D	$\mathcal{M}_1$
Situation 8	SDiPCA–MSI	-	D	$\mathcal{M}_2$
Situation 9	SDiPCA	$\mathcal{M}_3$	E	$\mathcal{M}_3$
Situation 10	SDiPCA	-	E	$\mathcal{M}_1$
Situation 11	SDiPCA	-	E	$\mathcal{M}_2$
Situation 12	PCA	$\mathcal{M}_1$	F	$\mathcal{M}_1$
Situation 13	PCA–EWC	F + $\mathcal{M}_2$	G	$\mathcal{M}_2$
Situation 14	PCA–EWC	-	G	$\mathcal{M}_1$
Situation 15	PCA–EWC	G + $\mathcal{M}_3$	H	$\mathcal{M}_3$
Situation 16	PCA–EWC	-	H	$\mathcal{M}_1$
Situation 17	PCA–EWC	-	H	$\mathcal{M}_2$
Situation 18	RPCA	$\mathcal{M}_1$	I	$\mathcal{M}_1$
Situation 19	RPCA	I + $\mathcal{M}_2$	J	$\mathcal{M}_2$
Situation 20	RPCA	J + $\mathcal{M}_3$	L	$\mathcal{M}_3$
Situation 21	IMPPCA	$\mathcal{M}_1$, $\mathcal{M}_2$	M	$\mathcal{M}_1$
Situation 22	IMPPCA	-	M	$\mathcal{M}_2$
Situation 23	IMPPCA	$\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$	O	$\mathcal{M}_1$
Situation 24	IMPPCA	-	O	$\mathcal{M}_2$
Situation 25	IMPPCA	-	O	$\mathcal{M}_3$
Situation 26	MCVA	$\mathcal{M}_1$, $\mathcal{M}_2$	P	$\mathcal{M}_1$
Situation 27	MCVA	-	P	$\mathcal{M}_2$
Situation 28	MCVA	$\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$	Q	$\mathcal{M}_1$
Situation 29	MCVA	-	Q	$\mathcal{M}_2$
Situation 30	MCVA	-	Q	$\mathcal{M}_3$

the similar design procedure. For Situations 21 and 22, a model is established based on data from modes $\mathcal{M}_1$ and $\mathcal{M}_2$, thus delivering optimal performance for the learned modes. When a new mode $\mathcal{M}_3$ appears, sufficient data are collected and then the model is retrained based on data from three modes. However, there are various unknown modes in practical systems. Storing all data and retraining the model seems to be expensive and less efficient [32]. To be consistent with other approaches, we calculate the global statistics of MCVA similar to [26] instead of Bayesian inference probability.

B. Computational complexity

The computation focuses on Algorithm 2. The term *flam*, a compound operation including one addition and one multiplication, is utilized to reflect operation counts [15].

For mode $\mathcal{M}_K$, the computational complexity is described as follows. The calculation of $\tilde{\boldsymbol{X}}_K^T \boldsymbol{Z}_K$ needs $(N_K - s)m^2 s$ flam before iteration. For $\boldsymbol{w}$, the gradient (17) requires $m^3 s + 2m^2 + 4m$ flam. The objective (13) needs $m^2 s + ms + 5m + 2s$ flam for each iteration. Thus, the calculation of Algorithm 1 needs $2m^3 s + (4 + 2s)m^2 + (2s + 22)m + 4s + 10$ flam in total for each iteration. With regard to $\boldsymbol{\beta}$, the gradient (18) needs $m^2 s^2 + 4s$ flam for each iteration. The calculation of Algorithm 1 requires $2m^2 s^2 + (2m^2 + 2m + 22)s + (4m + 10)$ flam in total for each iteration. The step 4) in Algorithm 2 needs $2sm^3 + (2s^2 + 2s + 6)m^2 + (4s + 28)m + (28s + 32)$ flam. The calculation of step 5) in Algorithm 2 needs $3N_K m + N_K$ flam. Assume the objective function converges after k_{total} iterations, it requires $(2sm^3 + (2s^2 + 2s + 6)m^2 + (4s + 28)m + (28s + 32))k_{total} + (3N_K m + N_K + (N_K - s)m^2 s)l$ in total.

C. Virtues and potential limitations

To illustrate the proposed method comprehensively, virtues and potential limitations are outlined. The advantages are summarized as follows:

1) Different from most multiple–model schemes [12], [13], [23], [26], [27], a single model is updated based on the previously learned knowledge and the current data when a novel mode arrives. Besides, it delivers the forward and backward transfer ability for successive dynamic modes. The learned knowledge is preserved to monitor the previous modes, which may be valuable to enhance the performance for future relevant modes.
2) SDiPCA–MSI shares similar advantages with DiPCA, such as robustness to collinearity, interpretability, prediction performance, etc [7], [8].
3) The importance measure (11) is more accurate than (10) of SI. Besides, MSI has more choices about the initialization of optimization issues, which is relevant for acquiring the optimal model parameters. Compared with [35] and [15], the importance measure is estimated when the learning finishes, without decoupling the importance measure and the optimization issue.

Some potential limitations are discussed below:

1) Since the objective function is nonconvex, only the local optimal parameters are acquired [7].
2) SDiPCA-MSI requires the similarity among modes and may fail once the modes are especially diverse [29]. In this case, it is essential to build another monitoring model.
3) Similar to most multimode monitoring methods [15], [21], [22], a set of data should be collected before relearning, which may cause trouble to real–time monitoring. Briefly, the proposed method is not able to monitor novel modes without any relevant information. To our best knowledge, this issue may be inevitable only by data, which could be settled by data and prior knowledge.
4) The mode is identified by prior knowledge, which is generally available in industrial systems. It has been mentioned that new modes and faults are indistinguishable without prior data from their modes [10]. Thus, it is more reliable to identify modes by data and expert experience.

V. Case studies

A. CSTH

The CSTH process is a nonlinear dynamic process and serves as a preferred benchmark for multimode process monitoring [10], [14], [39]. The CSTH model was built and relevant introduction was presented in [40]. The CSTH process mixes the hot water and cold water well to satisfy the demand. Three critical variables, namely, level, temperature and flow, are manipulated by PI controllers. Six related variables are selected for monitoring in this paper.

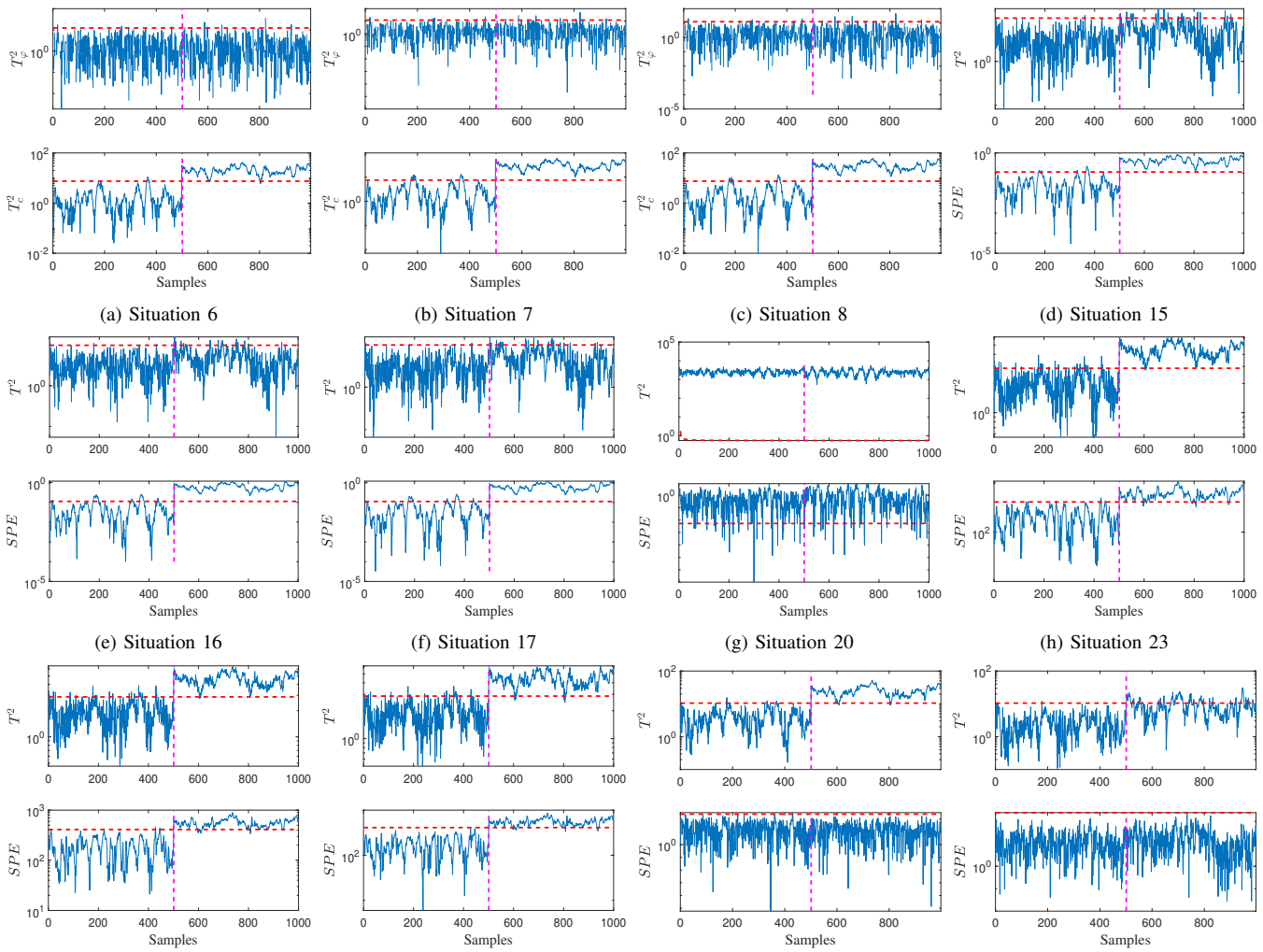
The comparative experiments have been designed in Table II. Two cases are considered and the operating settings are listed in Table III. For each case, three modes arrive sequentially. For each mode, 1000 normal samples are collected and 1000 testing samples are generated as follows:

1) *Case 1*: the level is added by 0.04 from 501st sample;
2) *Case 2*: the temperature is added by 0.055 from 501st sample.

For case 1, SDiPCA–MSI enables to monitor successive modes accurately. When two or more modes appear, the model is updated by consolidating new information while preserving the previously learned knowledge. Specifically, the FDRs of Situations 2 and 3 are 99% and 100%. The FDRs of Situations 6–8 are more than 93% and satisfactory. Meanwhile, the FARs are especially low. However, SDiPCA may fail to monitor multiple modes based on a single model. For instance, the FARs of Situations 5 and 11 are 18.40% and 53.20%. The learned experience of previous modes may be overwritten when a new mode is learned, which leads to abrupt performance decrease. PCA–EWC monitors multiple modes

TABLE III
Normal operating modes of CSTH

Case number	Mode label	Level SP	Temperature SP	Hot water valve
Case 1	$\mathcal{M}_1$	9	10.5	4.5
	$\mathcal{M}_2$	12	8	4
	$\mathcal{M}_3$	12	10.5	5.5
Case 2	$\mathcal{M}_1$	13	12	5.5
	$\mathcal{M}_2$	12	11	5
	$\mathcal{M}_3$	12	13.5	6



(a) Situation 6 (b) Situation 7 (c) Situation 8 (d) Situation 15

(e) Situation 16 (f) Situation 17 (g) Situation 20 (h) Situation 23

(i) Situation 24 (j) Situation 25 (k) Situation 28 (l) Situation 29

Fig. 2. Monitoring charts of case 2

based on a single mode, but the performance is unconvincing. For Situation 13, the FDR is 74.40% and the monitoring model cannot detect the fault accurately. For Situation 17, the FAR is 51%, which means that partial faulty samples are mistaken for normal samples. For Situations 18–20, RPCA model is updated based on the successive data, but it is not able to track the rapid changes in the entire dataset. The normal changes are mistaken for faults and the FARs are basically 100%. With regard to IMPPCA and MCVA, the faults can be detected precisely and timely. According to the scheme designed in Table II, IMPPCA and MCVA monitor the learned modes, and the performance should be excellent. The simulation results are summarized in Table IV. Partial monitoring charts are shown in Figure 1 in the supplementary material owing to space limitation.

For case 2, the FDRs of Situations 2 and 3 approach 100%. The model D trained by SDiPCA–MSI is capable of monitoring three modes and the FDRs are favorable. Although the model C is able to monitor two modes, the model E fails to detect three faults simultaneously. This phenomenon reveals the catastrophic forgetting issue of SDiPCA, where the information of previous modes is overlapped by new features. Moreover, the FDR of Situation 6 is 99.00% while the FDR of Situation 9 is 66.06%. It indicates that SDiPCA–MSI furnishes the forward transfer ability for future relevant modes, where the learned knowledge of previous modes may be beneficial for enhancing the monitoring performance of new modes. Briefly, the simulation results of Situations 1–11 illustrate the continual learning ability of SDiPCA–MSI for successive modes. For Situations 12–17, the FDRs of PCA–EWC are excellent, but the FARs are a little higher than SDiPCA–MSI. Similar to case 1, RPCA fails to monitor the multiple modes accurately. IMPPCA performs notably on case 2 and the FARs are less than 12.20%. MCVA is unable to supply prominent performance. Specifically, the FDR and FAR of Situation 27 are 29.52% and 28.71%, which signifies that normal data and faulty samples cannot be separated. Besides, the FDR of Situation 29 is 30.92%. Partial monitoring charts are depicted in Fig. 2 and more charts are illustrated in Figure 2 in the supplementary material.

In conclusion, SDiPCA–MSI alleviates the catastrophic forgetting issue of SDiPCA. The model is updated continually

TABLE IV
FDRs (%) AND FARs (%) FOR CSTH AND THE COAL PULVERIZING SYSTEM

Case number	Methods	CSTH				The coal pulverizing system					
		Case 1		Case 2		Case 3		Case 4		Case 5	
Indexes		FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR
Situation 1	SDiPCA	100	9.60	99.00	0.80	99.82	0	94.12	3.76	100	0.37
Situation 2	SDiPCA–MSI	99.00	0.40	98.39	1.41	100	0.48	98.16	0	100	0.18
Situation 3	SDiPCA–MSI	100	3.60	99.20	1.20	99.82	0.56	99.11	7.52	100	0.63
Situation 4	SDiPCA	100	6.00	98.59	1.40	100	0.64	98.16	0	100	0.09
Situation 5	SDiPCA	100	18.40	99.20	1.00	100	38.16	98.76	9.02	100	55.13
Situation 6	SDiPCA–MSI	93.37	1.80	99.00	2.60	86.09	0	100	5.96	99.78	0.10
Situation 7	SDiPCA–MSI	99.40	0.40	100	8.60	99.82	0.19	96.74	6.02	100	2.62
Situation 8	SDiPCA–MSI	94.38	4.60	100	8.80	99.88	5.28	98.49	0	100	0.45
Situation 9	SDiPCA	100	15.00	66.06	0.60	86.49	0	100	9.44	100	0.31
Situation 10	SDiPCA	100	2.80	92.87	0.40	100	59.77	99.90	27.07	100	72.88
Situation 11	SDiPCA	100	53.20	93.17	0.40	100	68.80	98.66	1.68	100	15.64
Situation 12	PCA	99.20	0	100	6.40	99.45	0	93.69	0	100	0.37
Situation 13	PCA–EWC	74.40	4.60	100	7.40	100	6.24	99.17	32.77	99.71	13.55
Situation 14	PCA–EWC	97.40	7.60	100	5.20	99.64	20.30	99.65	14.29	100	5.63
Situation 15	PCA–EWC	100	14.40	100	5.20	91.31	0	100	1.49	99.34	0.71
Situation 16	PCA–EWC	100	2.20	100	11.20	99.45	15.41	98.82	1.50	100	4.88
Situation 17	PCA–EWC	100	51.00	100	11.20	100	2.72	98.67	27.73	99.71	11.82
Situation 18	RPCA	100	99.00	100	99.80	100	100	100	100	100	100
Situation 19	RPCA	100	100	100	100	100	100	100	100	100	100
Situation 20	RPCA	100	100	100	100	100	100	100	100	100	100
Situation 21	IMPPCA	100	0.80	100	5.80	99.45	0	94.33	0	100	0.37
Situation 22	IMPPCA	100	4.20	100	12.20	100	35.84	97.67	0	99.71	15.27
Situation 23	IMPPCA	100	3.00	100	8.00	99.45	0	94.57	6.02	100	1.75
Situation 24	IMPPCA	100	5.60	99.60	7.40	100	39.84	97.67	0	99.71	3.66
Situation 25	IMPPCA	100	5.60	98.20	6.00	87.95	0	100	11.06	99.56	18.11
Situation 26	MCVA	100	3.20	97.79	5.20	100	43.61	96.16	0	100	1.12
Situation 27	MCVA	100	0.60	29.52	28.71	90.14	10.34	100	57.89	52.91	1.12
Situation 28	MCVA	100	1.60	98.59	1.60	100	27.44	98.57	27.82	100	0.63
Situation 29	MCVA	100	0.80	30.92	0.20	100	4.64	97.67	0	100	3.45
Situation 30	MCVA	100	9.80	94.38	6.60	88.12	0	100	26.96	99.78	0.51

based on the learned knowledge and new data when a new mode arrives, thus delivering optimal performance for successive modes. Since similarity exists among different modes, the learned knowledge of previous modes may contribute to building an accurate monitoring model for future new modes. PCA–EWC is also an effective algorithm with continual learning ability for multimode processes, where the data are required to be stationary and Gaussian distributed in each mode. Thus, the performance is worse than the proposed method in multimode dynamic systems. Moreover, the importance measure of SDiPCA–MSI is easier to estimate than PCA–EWC. As a typical adaptive method, RPCA is capable of dealing with slow normal variations but fails to track the dramatic changes in the entire dataset. IMPPCA provides the superior effect for this CSTH case. MCVA is unsatisfactory in some situations. For IMPPCA and MCVA, the model needs to be retrained from scratch when a new mode appears. They require considerable computation and storage resources for numerous modes, which makes it inappropriate for practical industrial systems.

B. The coal pulverizing system

In this paper, one unit of the 1000-MW ultra–supercritical thermal power plant is adopted to illustrate the effectiveness of SDiPCA–MSI, namely, the coal pulverizing system. The thermal power plant locates at Zhoushan, Zhejiang Province, China. The coal pulverizing system includes coal feeder, coal mill, rotary separator, raw coal hopper and stone coal scuttle, as shown in Fig. 3. The product of the coal pulverizing system, namely, the pulverized coal, should be provided with optimal temperature and coal fineness.

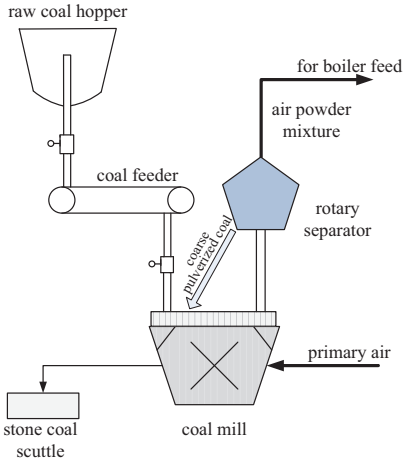


Fig. 3. Schematic diagram of the coal pulverizing system

This paper investigates the abnormalities from outlet temperature (case 3), rotary separators (case 4) and the coal feeders (case 5). To reduce false alarms and enhance detection accuracy, the variables are selected based on the professional knowledge and practical experience. The data information is summarized in Table V. For notation simplification, the

TABLE V
EXPERIMENTAL DATA OF THE PRACTICAL COAL PULVERIZING SYSTEM

Case number	Key variables	Mode number	NoTrS	NoTeS	Fault location	Coal type	Fault cause
Case 3	9 variables: pressure of air powder mixture, outlet temperature, primary air pressure and temperature, etc.	$\mathcal{M}_1$	1080	1080	533	Yinni	Hot primary air electric damper failure
		$\mathcal{M}_2$	1440	1440	626	Aomeng	Air leakage at primary air interface
		$\mathcal{M}_3$	1440	2880	384	Aomeng	Fluctuation of hot primary air control valve
Case 4	9 variables: rotary separator speed and current, coal feeding capacity, bearing temperature, etc.	$\mathcal{M}_1$	2880	2160	134	Aomeng	Large vibration
		$\mathcal{M}_2$	2880	720	120	Aomei	Cooling fan trip of inverter cabinet
		$\mathcal{M}_3$	2880	1080	806	Aomeng	Frequency conversion cabinet short circuit
Case 5	14 variables: current and speed of coal feeder, rotary separator speed and current, coal feeding capacity, etc.	$\mathcal{M}_1$	2520	1440	801	Aomeng	The coal feeder belt is broken
		$\mathcal{M}_2$	2160	1440	1101	Shenhun	Coal block of the coal pipe
		$\mathcal{M}_3$	2160	1440	984	Aomeng	The coal feeder does not drop coal

numbers of training and testing samples are denoted as NoTrS and NoTeS, respectively. The coal type switching is adopted as the basis for mode transformation, which could be estimated by the instantaneous coal feed and the total volume of coal. Since the coal plan is designed in advance based on market demand and characteristics of coal, it is feasible to get the coal consumption information in real time. Intuitively, the mode information is available for this case. The monitoring results of three cases are summarized in Table IV.

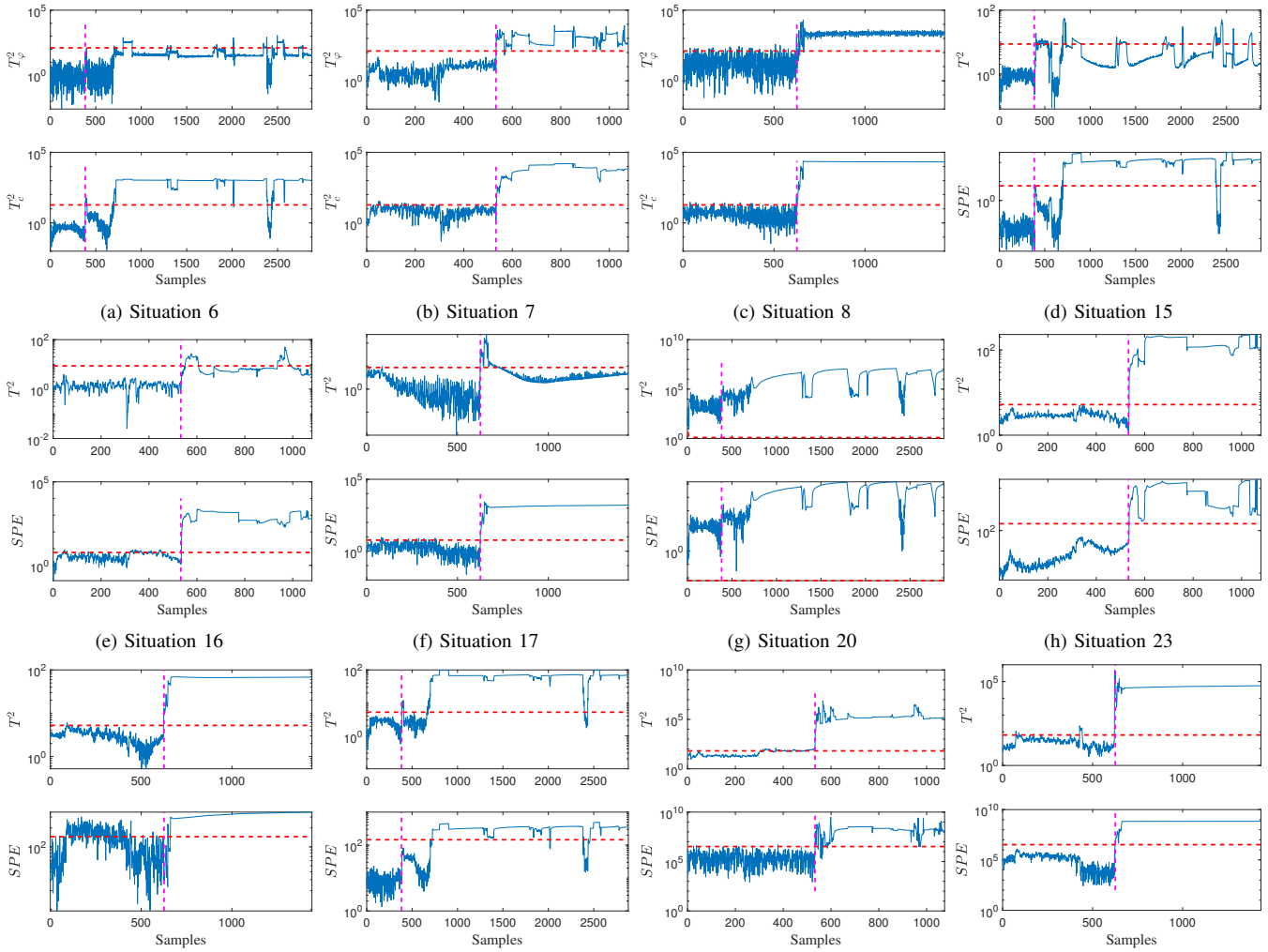
For case 3, partial monitoring charts are depicted in Fig. 4 and more charts are listed in Figure 3 in the supplementary material. SDiPCA–MSI can monitor two modes accurately and the FDRs are higher than 99% for Situations 2 and 3. However, the FAR of Situation 5 is 38.16%, which indicates that the model C of mode $\mathcal{M}_2$ fails to detect the fault in mode $\mathcal{M}_1$. Similar monitoring performance occurs in three successive modes. The FARs of Situations 10 and 11 are higher than 59%, which implies that the model E of mode $\mathcal{M}_3$ can not distinguish the real fault from normal samples in other modes. For Situations 6–8, SDiPCA–MSI is able to monitor three modes simultaneously, and the performance is comparable with Situations 1, 4 and 9. Actually, the significant information from three modes is preserved in model D, which is sufficient to deliver superior performance. For PCA–EWC, the FDRs are more than 90%, but the FARs of Situations 14 and 16 are 20.30% and 15.41%, respectively. Moreover, RPCA is not able to monitor the multimode processes and the FARs are 100%. For IMPPCA, the FARs of Situations 22 and 24 are higher than 35%. For MCVA, the FARs of Situations 26 and 28 are 43.61% and 27.44%, respectively. Although the model is retrained based on all mode data, two methods still fail to monitor multiple modes accurately.

The aforementioned analysis also applies to cases 4 and 5. For case 4, the detection results of SDiPCA–MSI and SDiPCA are excellent except for Situation 10, where the FAR is 27.07% and the catastrophic forgetting of SDiPCA is reflected. Besides, the FARs of Situation 6 and Situation 9 are 5.96% and 9.44%, which implies that the previously learned knowledge contributes to enhancing the monitoring performance of future new mode $\mathcal{M}_3$. For PCA–EWC, the FARs of Situations 13, 14 and 17 are relatively high. That is, SDiPCA–MSI is superior to PCA–EWC. IMPPCA performs excellently and only the FAR of Situation 25 is more than 10%. For MCVA, the FARs of Situations 27, 28 and 30 are higher than 26%. Similarly, for case 5, the FARs of Situations 5, 10 and 11 are high, which indicate that the information of previously learned modes is overlapped by new data. SDiPCA–MSI outperforms other methods, where the FDRs approach 100% and the FARs are lower than 3%. For PCA–EWC, the FARs of Situations 13 and 17 are higher than 10%. Besides, RPCA is not able to monitor the process accurately and the FARs are 100%. For IMPPCA, the FARs of Situations 22 and 25 are higher than 15%. For MCVA, the FDR of Situation 27 is 52.91%. Owing to space limitation, the monitoring charts of cases 4 and 5 are illustrated in Figures 4 and 5 in the supplementary material.

Overall, SDiPCA–MSI delivers the most advantageous performance for successive modes among five methods. The previously learned knowledge is consolidated without storing training samples while assimilating information from new modes, thus avoiding abrupt performance degradation for previous modes. When similar modes recur, the proposed method can build an accurate model based on limited new data because partial significant information has already been preserved in the learned model. Although PCA–EWC furnishes continual learning ability for multimode processes, the monitoring performance is not satisfactory because it requires that data are stationary and Gaussian distributed in each mode. RPCA fails to monitor the modes and track the process adaptively. As the representatives of multiple–model schemes, IMPPCA and MCVA cannot monitor multiple modes accurately. In addition, the model needs to be retrained on all normal data when a new mode appears and all data are required to be stored. In terms of detection accuracy, storage and computation resources, SDiPCA–MSI provides optimal performance for industrial systems.

VI. CONCLUSION

This paper presented a novel extension of DiPCA with continual learning ability for multimode dynamic process monitoring, where sparse representation is adopted to enhance model interpretability and modified synaptic intelligence is developed to measure the parameter importance. The proposed SDiPCA–MSI method extracts the significant information of new modes while retaining the previously learned knowledge simultaneously, thus avoiding abrupt performance decrease for the learned modes. Different from traditional multimode processes, data from all modes are not required to be available



(a) Situation 6 (b) Situation 7 (c) Situation 8 (d) Situation 15

(e) Situation 16 (f) Situation 17 (g) Situation 20 (h) Situation 23

(i) Situation 24 (j) Situation 25 (k) Situation 28 (l) Situation 29

Fig. 4. Monitoring charts of case 3

before learning and the model enables to adapt to the changing modes. Besides, virtues and potential limitations are analyzed to understand the proposed algorithm thoroughly. Although the mode identification depends on prior knowledge, it is reliable and convenient because the relevant information has already existed during the operating process. Compared with several typical multimode algorithms, the effectiveness and superiorities of the proposed method have been illustrated by a CSTH case and a practical coal pulverizing system.

In future, we would investigate the automatic mode identification based on data and prior knowledge for multimode dynamic processes. Besides, the novel modes with little information would be studied for online monitoring.

APPENDIX

A. Deviation of importance measure

Assuming that the surrogate loss for each parameter up to Kth mode, $J_{total}^{K}\left(\theta_i\right)$, is defined by a quadratic function. It can be decomposed into two terms as

$$J_{total}^{K}\left(\theta_i\right) \equiv \hat{\varpi}_i^K\left(\theta_i-\hat{\theta}_i^K\right)^2=\tilde{J}_K\left(\theta_i\right)+\gamma_K J_{total}^{K-1}\left(\theta_i\right) \tag{34}$$

where $\tilde{J}_K\left(\theta_i\right)$ is the loss for the current Kth mode, $J_{total}^{K-1}\left(\theta_i\right)$ is the surrogate loss for previous $(K-1)$ modes.

1) Synaptic intelligence [34]: The importance measure $\hat{\varpi}_i^K$ is calculated by

$$\begin{aligned}
\hat{\varpi}_i^K &= \frac{\tilde{J}_K\left(\theta_i\right)+\gamma_K J_{total}^{K-1}\left(\theta_i\right)}{\left(\theta_i-\hat{\theta}_i^K\right)^2} \\
&\underline{\underline{\theta_i=\hat{\theta}_i^{K-1}}}\ \frac{\tilde{J}_K\left(\hat{\theta}_i^{K-1}\right)}{\left(\hat{\theta}_i^{K-1}-\hat{\theta}_i^K\right)^2} \\
&= \frac{\tilde{J}_K\left(\hat{\theta}_i^{K-1}\right)-\tilde{J}_K\left(\hat{\theta}_i^K\right)}{\left(\hat{\theta}_i^{K-1}-\hat{\theta}_i^K\right)^2}+\frac{\tilde{J}_K\left(\hat{\theta}_i^K\right)\cdot\hat{\varpi}_i^{K-1}}{J_{total}^{K-1}\left(\hat{\theta}_i^K\right)} \\
&= \frac{\varpi_i^K}{\left(\hat{\theta}_i^{K-1}-\hat{\theta}_i^K\right)^2}+\frac{\tilde{J}_K\left(\hat{\theta}_i^K\right)\cdot\hat{\varpi}_i^{K-1}}{J_{total}^{K-1}\left(\hat{\theta}_i^K\right)} \\
&\approx \frac{\varpi_i^K}{\left(\hat{\theta}_i^{K-1}-\hat{\theta}_i^K\right)^2}+\hat{\varpi}_i^{K-1}
\end{aligned} \tag{35}$$

where ϖ_i^K is the difference between the losses $\tilde{J}_K(\theta_i)$ in (7) for the only Kth mode before and after training, and calculated by (8). Note that $J_{total}^{K-1}\left(\hat{\theta}_i^K\right) = \left(\hat{\theta}_i^{K-1} - \hat{\theta}_i^K\right)^2 \cdot \hat{\varpi}_i^{K-1}$, $J_{total}^{K-1}\left(\hat{\theta}_i^{K-1}\right) = 0$. Since $\tilde{J}_K\left(\hat{\theta}_i^K\right) \neq J_{total}^{K-1}\left(\hat{\theta}_i^K\right)$ in most cases, the estimation (35) of importance measure is inaccurate.

2) Modified synaptic intelligence: For MSI, the importance measure $\hat{\varpi}_i^K$ is calculated by

$$\begin{aligned}\hat{\varpi}_i^K &= \frac{J_{total}^K(\theta_i)}{\left(\theta_i - \hat{\theta}_i^K\right)^2} \\ &\underline{\underline{\theta_i = \theta_i^0}} \ \frac{J_{total}^K\left(\theta_i^0\right)}{\left(\theta_i^0 - \hat{\theta}_i^K\right)^2} \\ &= \frac{J_{total}^K\left(\theta_i^0\right) - \hat{\varpi}_i^K\left(\hat{\theta}_i^K - \hat{\theta}_i^K\right)^2}{\left(\theta_i^0 - \hat{\theta}_i^K\right)^2} \\ &= \frac{J_{total}^K\left(\theta_i^0\right) - J_{total}^K\left(\hat{\theta}_i^K\right)}{\left(\theta_i^0 - \hat{\theta}_i^K\right)^2} \\ &= \frac{\varpi_i^K}{\left(\theta_i^0 - \hat{\theta}_i^K\right)^2}\end{aligned} \tag{36}$$

where the initial setting θ_i^0 is random but satisfies the constraints. According to (34), $J_{total}^K\left(\hat{\theta}_i^K\right) = 0$, ϖ_i^K is calculated by (8) and the loss function is $J_{total}^K(\theta_i)$ in (9). Obviously, (36) is more accurate than (35).

When $\theta_i^0 = \hat{\theta}_i^{K-1}$, (36) can be reformulated as

$$\begin{aligned}\hat{\varpi}_i^K =& \frac{J_{total}^K\left(\hat{\theta}_i^{K-1}\right) - J_{total}^K\left(\hat{\theta}_i^K\right)}{\left(\hat{\theta}_i^{K-1} - \hat{\theta}_i^K\right)^2} \\ =& \frac{\varpi_i^K}{\left(\hat{\theta}_i^{K-1} - \hat{\theta}_i^K\right)^2}\end{aligned} \tag{37}$$

where ϖ_i^K is the difference between the losses $J_{total}^K(\theta_i)$ for all modes before and after training. Although (37) and (35) share the same initial setting, (35) is an approximated value and generally inaccurate. That is to say, (37) is more accurate than (35).

B. Proximal gradient descent

For a nonsmooth objective function, it is decomposed into:

$$f(\boldsymbol{\theta}) = g(\boldsymbol{\theta}) + h(\boldsymbol{\theta})$$

where $g(\boldsymbol{\theta})$ is differential but $h(\boldsymbol{\theta})$ is nondifferentiable. The proximal function is defined as

$$\begin{aligned}\boldsymbol{\theta}^+ =& \arg\min_{\boldsymbol{z}} \frac{1}{2t}\|\boldsymbol{z} - (\boldsymbol{\theta} - t\nabla g(\boldsymbol{\theta}))\|_2^2 + h(\boldsymbol{z}) \\ :=& prox_{h,t}(\boldsymbol{\theta} - t\nabla g(\boldsymbol{\theta}))\end{aligned}$$

where the proximal function $prox$ is defined as [36]

$$prox_{h,t}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{z}} \frac{1}{2t}\|\boldsymbol{z} - \boldsymbol{\theta}\|_2^2 + h(\boldsymbol{z})$$

If $h(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_1$, the proximal gradient can be calculated by the soft threshold

$$\begin{aligned}prox_{h,t}(\boldsymbol{\theta}) =& \arg\min_{\boldsymbol{z}} \frac{1}{2t}\|\boldsymbol{z} - \boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1 \\ =& S_{\lambda t}(\boldsymbol{\theta})\end{aligned} \tag{38}$$

where $S_{\lambda t}(\boldsymbol{\theta})$ has an analytical solution [37]:

$$[S_{\lambda t}]_i = \begin{cases}\theta_i - \lambda t, & \theta_i > \lambda t \\ 0, & |\theta_i| \leq \lambda t \\ \theta_i + \lambda t, & \theta_i < \lambda t\end{cases} \tag{39}$$

where t is the step size.

REFERENCES

[1] J. Shi, J. Sun, Y. Yang, and D. Zhou, "Distributed self-triggered formation control for multi-agent systems," *Science in China Series F: Information Sciences*, vol. 63, no. 10, pp. 1–3, 2020.

[2] L. Zhou, J. Zheng, Z. Ge, Z. Song, and S. Shan, "Multimode process monitoring based on switching autoregressive dynamic latent variable model," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8184–8194, 2018.

[3] P.-E. Odiowei and Y. Cao, "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 36–45, 2010.

[4] I. Jaffel, O. Taouali, M. F. Harkat, and H. Messaoud, "Moving window KPCA with reduced complexity for nonlinear dynamic process monitoring," *ISA Transactions*, vol. 64, pp. 184–192, 2016.

[5] K. E. S. Pilario and Y. Cao, "Canonical variate dissimilarity analysis for process incipient fault detection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5308–5315, 2018.

[6] R. Raveendran and B. Huang, "Two layered mixture Bayesian probabilistic PCA for dynamic process monitoring," *Journal of Process Control*, vol. 57, pp. 148–163, 2017.

[7] Y. Dong and S. J. Qin, "A novel dynamic PCA algorithm for dynamic data modeling and process monitoring," *Journal of Process Control*, vol. 67, pp. 1–11, 2018.

[8] S. J. Qin, Y. Dong, Q. Zhu, J. Wang, and Q. Liu, "Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring," *Annual Reviews in Control*, vol. 50, pp. 29–48, 2020.

[9] F. Lv, C. Wen, and M. Liu, "Representation learning based adaptive multimode process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 181, pp. 95–104, 2018.

[10] M. Quiñones-Grueiro, A. Prieto-Moreno, C. Verde, and O. Llanes-Santiago, "Data-driven monitoring of multimode continuous processes: A review," *Chemometrics and Intelligent Laboratory Systems*, vol. 189, pp. 56–71, 2019.

[11] H. Wu and J. Zhao, "Self-adaptive deep learning for multimode process monitoring," *Computers & Chemical Engineering*, vol. 141, p. 107024, 2020.

[12] W. Du, Y. Tian, and F. Qian, "Monitoring for nonlinear multiple modes process based on LL–SVDD–MRDA," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1133–1148, 2014.

[13] Y. Liu, J. Zeng, J. Bao, and L. Xie, "A unified probabilistic monitoring framework for multimode processes based on probabilistic linear discriminant analysis," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6291–6300, 2020.

[14] K. Huang, Y. Wu, C. Yang, G. Peng, and W. Shen, "Structure dictionary learning-based multimode process monitoring and its application to aluminum electrolysis process," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1989–2003, 2020.

[15] J. Zhang, D. Zhou, and M. Chen, "Monitoring multimode processes: a modified PCA algorithm with continual learning ability," *Journal of Process Control*, vol. 103, pp. 76–86, 2021.

[16] H. Ma, Y. Hu, and H. Shi, "A novel local neighborhood standardization strategy and its application in fault detection of multimode processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 118, pp. 287–300, 2012.

[17] X. Deng, N. Zhong, and L. Wang, "Nonlinear multimode industrial process fault detection using modified kernel principal component analysis," *IEEE Access*, vol. 5, pp. 23 121–23 132, 2017.

[18] C. Tong, A. Palazoglu, and X. Yan, "An adaptive multimode process monitoring strategy based on mode clustering and mode unfolding," *Journal of Process Control*, vol. 23, no. 10, pp. 1497–1507, 2013.

[19] L. M. Elshenawy, S. Yin, A. S. Naik, and S. X. Ding, "Efficient recursive principal component analysis algorithms for process monitoring," *Industrial & Engineering Chemistry Research*, vol. 49, no. 1, pp. 252–259, 2010.

[20] J. Chen and C. Zhao, "Exponential stationary subspace analysis for stationary feature analytics and adaptive nonstationary process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8345–8356, 2021.

[21] W. Yu and C. Zhao, "Recursive exponential slow feature analysis for fine-scale adaptive processes monitoring with comprehensive operation status identification," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3311–3323, 2019.

[22] W. Yu, C. Zhao, and B. Huang, "Recursive cointegration analytics for adaptive monitoring of nonstationary industrial processes with both static and dynamic variations," *Journal of Process Control*, vol. 92, pp. 319–332, 2020.

[23] W. Shao, Z. Ge, L. Yao, and Z. Song, "Bayesian nonlinear Gaussian mixture regression and its application to virtual sensing for multimode industrial processes," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 871–885, 2020.

[24] Q. Jiang and X. Yan, "Multimode process monitoring using variational Bayesian inference and canonical correlation analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1814–1824, 2019.

[25] X. Xie and H. Shi, "Dynamic multimode process modeling and monitoring using adaptive gaussian mixture models," *Industrial & Engineering Chemistry Research*, vol. 51, no. 15, pp. 5497–5505, 2012.

[26] J. Zhang, H. Chen, S. Chen, and X. Hong, "An improved mixture of probabilistic PCA for nonlinear data-driven process monitoring," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 198–210, 2019.

[27] Q. Wen, Z. Ge, and Z. Song, "Multimode dynamic process monitoring based on mixture canonical variate analysis model," *Industrial & Engineering Chemistry Research*, vol. 54, no. 5, pp. 1605–1614, 2015.

[28] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[29] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing change: Continual learning in deep neural networks," *Trends in Cognitive Sciences*, vol. 24, no. 12, pp. 2018–1040, 2020.

[30] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, and A. Grabska-Barwinska, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, no. 13, pp. 3521–3526, 2017.

[31] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[32] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Communications*, vol. 11, no. 1, pp. 4069–4069, 2020.

[33] N. Y. Masse, G. D. Grant, and D. J. Freedman, "Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 44, pp. E10 467–E10 475, 2018.

[34] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, vol. 70, 2017, pp. 3987–3995.

[35] D. Park, S. Hong, B. Han, and K. M. Lee, "Continual learning by asymmetric loss approximation with single-side overestimation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3335–3344.

[36] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, vol. 28, 2015, pp. 379–387.

[37] N. K. Dhingra, S. Z. Khong, and M. R. Jovanovic, "The proximal augmented Lagrangian method for nonsmooth composite optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2861–2868, 2019.

[38] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

[39] X. Xu, L. Xie, and S. Wang, "Multimode process monitoring with PCA mixture model," *Computers & Electrical Engineering*, vol. 40, no. 7, pp. 2101–2112, 2014.

[40] N. F. Thornhill, S. C. Patwardhan, and S. L. Shah, "A continuous stirred tank heater simulation model with applications," *Journal of Process Control*, vol. 18, no. 3, pp. 347–360, 2008.

Jingxin Zhang received B.E. degree from School of Electrical Engineering and Automation, Harbin Engineering University, Harbin, China, the M.E. degree from Control Science and Engineering, Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively. Since 2018, she has been pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China. Her research interests are data-driven fault detection and diagnosis, performance monitoring and their applications in the industrial process.

Donghua Zhou (SM'99-F'19, IEEE) received the B.Eng., M. Sci., and Ph.D. degrees all in electrical engineering from Shanghai Jiaotong University, China, in 1985, 1988, and 1990, respectively. He was an Alexander von Humboldt research fellow with the university of Duisburg, Germany from 1995 to 1996, and a visiting scholar with Yale university, USA from 2001 to 2002. He joined Tsinghua university in 1996, and was promoted as full professor in 1997, he was the head of the department of automation, Tsinghua university, during 2008 and 2015. He is now a vice president, Shandong University of Science and Technology, and a joint professor of Tsinghua university. He has authored and coauthored over 230 peer-reviewed international journal papers and 7 monographs in the areas of fault diagnosis, fault-tolerant control and operational safety evaluation. Dr. Zhou is a fellow of IEEE, CAA and IET, a member of IFAC TC on SAFEPROCESS, an associate editor of Journal of Process Control, the vice Chairman of Chinese Association of Automation (CAA), the TC Chair of the SAFEPROCESS committee, CAA. He was also the NOC Chair of the 6th IFAC Symposium on SAFEPROCESS 2006.

Maoyin Chen received the B.S. degree in mathematics and the M.S. degree in control theory and control engineering from Qufu Normal University, Shandong, China, in 1997 and 2000, respectively, and the Ph.D. degree in control theory and control engineering from Shanghai Jiaotong University, Shanghai, China, in 2003. From 2003 to 2005, he was a Postdoctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China. From 2006 to 2008, he visited Potsdam University, Potsdam, Germany, as an Alexander von Humboldt Research Fellow. Since October 2008, he has been an Associated Professor with the Department of Automation, Tsinghua University. He has authored and coauthored over 110 peer-reviewed international journal papers. He has won the first prize in natural science (2011, ranked first) and the second prize (2019, ranked first) of CAA. His research interests include fault prognosis and complex systems.

Xia Hong received the B.Sc. and M.Sc. degrees from the National University of Defense Technology, China, in 1984 and 1987, respectively, and the Ph.D. degree from The University of Sheffield, U.K., in 1998, all in automatic control. She was a Research Assistant with the Beijing Institute of Systems Engineering, Beijing, China, from 1987 to 1993. She was a Research Fellow with the Department of Electronics and Computer Science, University of Southampton, from 1997 to 2001.

She is currently a Professor with the Department of Computer Science, School of Mathematical, Physical and Computational Sciences, University of Reading. She is actively involved in research into nonlinear systems identification, data modeling, estimation and intelligent control, neural networks, pattern recognition, learning theory, and their applications. She has authored over 170 research papers, and co-authored a research book. Dr. Hong received the Donald Julius Groen Prize from IMechE in 1999.