Monitoring multimode nonlinear dynamic processes: an efficient sparse dynamic approach with continual learning ability

Monitoring multimode nonlinear dynamic processes: an efficient sparse dynamic approach with continual learning ability

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

***Abstract*— Industrial processes generally operate under multiple modes and a global monitoring approach, built upon combining local models which are aimed at each mode, requires complete data from all potential modes to be available. However, practical data are generated and collected in a steady stream, which makes it difficult if not impossible to process. This paper proposes an efficient sparse dynamic inner principal component analysis algorithm for multimode nonlinear dynamic process monitoring, which aims to build a single monitoring model with continual learning ability for successive modes. To reduce the storage and computational costs, only a few representative data from each mode are selected based on cosine similarity and replayed for retraining when a new mode arrives, which are sufficient to reflect the operating condition of each mode. Inspired by replay continual learning, data from all existing modes are preprocessed by its own statistics and then regarded as a whole data set, followed by building a single multimode monitoring model. The multimode dynamic latent variables are extracted from data in raw format, via a vector autoregressive model. Therefore, the proposed method is not constrained by the mode similarity, which makes it appropriate for diverse modes and convenient for long-term monitoring tasks. Besides, the proposed method can deal with nonlinearity and a regularization term is added to avoid the potential overfitting issue. Compared with state-of-the-art multimode monitoring methods, the effectiveness of the proposed approach is demonstrated by continuous stirred tank heater and a practical industrial system.**

***Index Terms*— Multimode nonlinear dynamic process monitoring, sparse dynamic inner principal component analysis, replay continual learning, alternating direction method of multipliers**

I. Introduction

To enhance operational safety, data-driven process monitoring has attracted wide attention and acquired fruitful achievements [1]–[4]. Industrial systems are essentially dynamic with internal variables' relationship being nonlinear [5]–[7]. Therefore, there are many researches devoted to nonlinear/dynamic process monitoring [6]. Canonical variate analysis (CVA) established a state space model and was applied to nonlinear dynamic process monitoring [6]. Dynamic and static latent variables were extracted by dynamic inner principal component analysis (DiPCA), which was utilized to monitor linear dynamic processes [5].

Since industrial processes universally operate under multiple conditions, multimode process monitoring has experienced rapid development [1], [8], [9]. It has been mentioned in [1] that multiple-model methods are the mainstream, where local models are built for each mode correspondingly or a global model is designed based on a weighted sum of local models [10], [11]. For instance, mixture of CVA (MCVA) constructed local CVA models within the framework of Gaussian mixture model and was applied to multimode dynamic process monitoring [11]. An improved mixture of probabilistic principal component analysis (IMPPCA) [10] divided data into several clusters and was applied to multimode monitoring [12], where a global model was constructed via a weighted combination of local principal component analysis (PCA) models based on Bayesian inference. Moreover, a novel nonstationary discrete convolution kernel was proposed to characterize the multimodality behavior [13], to overcome the limitations of radial basis function kernel for multimode processes. Furthermore, a Dirichlet process Gaussian mixed model was investigated to identify the mode and nonlinear features were extracted based on t-distributed stochastic neighbor embedding, and then a monitoring index based on support vector data description was designed for comprehensive monitoring [14]. The approaches require complete data from all possible modes indicating that the model may need to be retrained from scratch if a new mode is encountered, thus consuming expensive storage and computing resources. However, diverse novel modes would appear continuously in future, and thus it is intractable to collect complete data before learning in industrial systems [1] and it may be unnecessary to build local models for each mode. Aimed at above issues, it is valuable and meaningful to investigate an efficient monitoring approach for multimode dynamic modes, where the model is learned over time and implemented. Data from previous modes are accumulated efficiently with limited storage resources.

Recently, continual learning has attracted wide attention and been applied to image processing, which extracts features

This work was supported by National Natural Science Foundation of China [grant numbers 61873143, 62203107,]. (Corresponding authors: Jingxin Zhang; Maoyin Chen)

Jingxin Zhang is with the School of Automation, Southeast University, Nanjing 210096, China (e-mail: zjx18@tsinghua.org.cn).

Maoyin Chen is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: mychen@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.

from an infinite stream of data and accumulates the acquired knowledge for future learning [15]–[17]. Therefore, continual learning may provide an efficient solution for sequential modes, where complete data are not required before learning. The major challenge is catastrophic forgetting, namely, the previously learned knowledge may be overwritten and the performance on a previous mode may degrade over time significantly when new modes arrive. There are diverse techniques on continual learning [16], [17], including regularization [18], manipulating data memory replay [19] and parameter isolation [20]. Regularization-based continual learning has already been utilized for multimode process monitoring [12], [21]–[23], where a single model is updated continuously by extracting new features while preserving the previously learned knowledge simultaneously. More specifically, a regularization term is designed to make the parameters close to the previous ones and one key point is to evaluate the importance of model parameters accurately. For instance, a modified PCA was proposed in [21] and denoted as PCA–EWC, where the importance was measured by elastic weight consolidation (EWC) [18]. However, PCA–EWC requires that data follow Gaussian distribution and it is generally intractable to calculate the Fisher information matrix accurately. To overcome this drawback, a novel spare PCA was presented and synaptic intelligence (SI) was adopted to evaluate the parameter importance [22]. Besides, sparse representation was utilized to enhance model interpretability and alleviate the catastrophic forgetting issue further [15]. To settle dynamics of data, a sparse DiPCA (SDiPCA) with continual learning ability was presented [12], where modified SI (MSI) was investigated to estimate the sensitivity of parameter changes to the loss. This method was referred to as SDiPCA–MSI. However, the performance would decrease suddenly when the modes are diverse, which makes it only appropriate for short-term monitoring tasks [15]. Besides, PCA–EWC and SDiPCA–MSI ignore the nonlinear relationship among data. In practical systems, various modes would appear in a steady stream, where variables are dynamic and nonlinear. Therefore, it is essential to investigate an efficient method to deal with diverse successive modes, with limited computing and storage resources.

Against this background, this paper investigates an efficient nonlinear dynamic monitoring method for successive modes, where data are generated and collected sequentially. First, a nonlinear sparse DiPCA (NSDiPCA) algorithm is presented to monitor a single nonlinear dynamic mode, where data are mapped to a high-dimensional feature space and a regularization term is introduced to avoid the potential overfitting issue. Then, motivated by replay continual learning, an efficient multimode NSDiPCA (MNSDiPCA) is proposed for sequential modes, where representative data from each mode are stored and combined with the current mode data for future training when a new mode appears. Since significant features are extracted from multimode data in raw format, the proposed method can monitor the current and previous modes accurately, without the requirement of mode similarity.

The contributions of this paper are summarized below:

a) This paper proposes an efficient MNSDiPCA for multimode nonlinear dynamic process monitoring, where data from diverse modes are collected over time and the model is retrained when a new mode arrives.
b) Compared with traditional multiple-model methods [1], [10], [11], data from all possible modes are not required and only a few representative data are stored instead of all samples, which are preprocessed by its statistics and then regarded as a whole dataset. In addition, a single model is established for sequential modes without the requirement of mode identification.
c) Compared with current multimode monitoring methods with continual learning ability [12], [21], [22], partial data from previous modes are stored and significant features are extracted from original data. Therefore, the proposed method is free from mode similarity, which allows it to monitor diverse modes and makes it appropriate for long-term monitoring tasks.

The rest of this paper is organized as follows. Section II introduces NSDiPCA for a single mode briefly and describes the multimode problem. Section III describes the core of MNSDiPCA for sequential modes and settles the optimization problem by alternating direction method of multipliers (ADMM). In addition, the off-line training and online monitoring phases are summarized. Continuous stirred tank heater (CSTH) and a practical industrial system are employed to demonstrate the effectiveness of the proposed method in Section IV. Section V is devoted to the conclusion.

II. Preliminary

A. The system model for single mode nonlinear dynamic process

The proposed NSDiPCA seeks to extract dynamic latent variables from observational data that contain most dynamic variations. Let observational data set be denoted as $\boldsymbol{X} = \{\boldsymbol{x}_k\}$, $k = 1, \cdots, N$, as time instance. N is the number of samples and $\boldsymbol{x} \in R^m$ is a sample vector of m variables.

Generally, a prelearning step is conducted before training, which can be represented by a mapping function $\mathcal{F}$: $\boldsymbol{x} \longmapsto \phi(\boldsymbol{x})$, and $\phi(\boldsymbol{x}) \in R^M$. There are many choices about ϕ and this step can also be regarded as unsupervised learning. For instance, ϕ can be represented by autoencoder to filter out noise or thin plate spline [24] to deal with nonlinearity. Similar to [25], the following mapping function $\phi(\boldsymbol{x})$ is adopted to cope with nonlinearity in this paper.

$$\phi(\boldsymbol{x}) = [x_m^2, \cdots, x_1^2, x_m x_{m-1}, \cdots, x_m x_1, x_{m-1}x_{m-2}, \cdots, x_{m-1}x_1, \cdots, x_2 x_1, x_m, \cdots, x_1] \in R^M \tag{1}$$

where $M = \frac{m^2+3m}{2}$, and x_i is the ith dimension of $\boldsymbol{x}$, $i = 1, \cdots, m$. For convenience, $\phi(\boldsymbol{x})$ is denoted as $\boldsymbol{x}_\phi$. The high-dimensional data are denoted as $\boldsymbol{X}_\phi \in R^{N\times M}$ by using (1) and the kth sample is denoted as $\boldsymbol{x}_{\phi,k}$ correspondingly.

Similar to DiPCA, NSDiPCA establishes a vector autoregressive (VAR) model to extract the most predictable information and characterize the dynamic relationship. The latent variables are defined as

$$t_k = \boldsymbol{x}_{\phi,k}^T \boldsymbol{w} \tag{2}$$

where $\boldsymbol{w} \in R^M$ is the weight vector with the constraint $\|\boldsymbol{w}\|_2 = 1$. Meanwhile, t_k could also be depicted by a linear combination of the past ones, namely,

$$t_k = \sum_{j=1}^{s} \beta_i t_{k-j} + r_k \tag{3}$$

where r_k is the Gaussian noise at kth instant, and s is the autoregressive order. Let $\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_s]^T$, $\|\boldsymbol{\beta}\|_2 = 1$. According to (2) and (3), the prediction of dynamic latent variables is reformulated as:

$$\begin{aligned}\hat{t}_k &= \sum_{j=1}^{s} \boldsymbol{x}_{\phi,k-j}^T \boldsymbol{w} \beta_j \\ &= \left[\boldsymbol{x}_{\phi,k-1}^T \ \cdots \ \boldsymbol{x}_{\phi,k-s}^T\right] (\boldsymbol{\beta} \otimes \boldsymbol{w})\end{aligned}$$

where $\otimes$ denotes the Kronecker product. The covariance between t_k and $\hat{t}_k$ is maximized to extract the dynamic information, namely,

$$\sum_{k=s+1}^{N} \boldsymbol{w}^T \boldsymbol{x}_{\phi,k} \left[\boldsymbol{x}_{\phi,k-1}^T \ \cdots \ \boldsymbol{x}_{\phi,k-s}^T\right] (\boldsymbol{\beta} \otimes \boldsymbol{w}) \tag{4}$$

Construct the matrices based on $\boldsymbol{X}_\phi$,

$$\boldsymbol{X}_\phi^{(j)} = [\boldsymbol{x}_{\phi,j} \ \ \boldsymbol{x}_{\phi,j+1} \ \cdots \ \boldsymbol{x}_{\phi,N-s+j-1}]^T, \ \ j = 1, \cdots, s+1 \tag{5}$$

$$\boldsymbol{Z} = \left[\boldsymbol{X}_\phi^{(1)} \ \ \boldsymbol{X}_\phi^{(2)} \ \cdots \ \boldsymbol{X}_\phi^{(s)}\right] \tag{6}$$

To avoid potential overfitting in (4) and alleviate catastrophic forgetting [15], the objective of NSDiPCA is reformulated as

$$\begin{aligned}&\min \ J(\boldsymbol{w}, \boldsymbol{\beta}) = -\boldsymbol{w}^T \left(\boldsymbol{X}_\phi^{(s+1)}\right)^T \boldsymbol{Z} (\boldsymbol{\beta} \otimes \boldsymbol{w}) + \lambda_1 \boldsymbol{\beta}^T \boldsymbol{D} \boldsymbol{\beta} \\ &s.t. \ \ \boldsymbol{w}^T \boldsymbol{w} = 1, \ \ \boldsymbol{\beta}^T \boldsymbol{\beta} = 1\end{aligned} \tag{7}$$

where $\boldsymbol{D}$ is a weighting matrix to make $\boldsymbol{\beta}$ sparse and λ_1 is a predefined regularization coefficient.

Define a recursively reduced data set from $\boldsymbol{X}_\phi$ and extract l dynamic latent variables successively by optimizing the objective (7). The estimation of s and l can refer to [5].

B. Problem statement

Define multiple modes $\mathcal{M}_K$, $K = 1, 2, \cdots$. Let observational data be denoted as $\boldsymbol{X}_K^0 \in R^{N_K \times m}$ and N_K is the number of samples in each mode $\mathcal{M}_K$. Traditional multimode process monitoring methods [9]–[11] need to store all normal data from each mode, which would consume expensive storage resources. To our best knowledge, massive industrial data may contain limited effective information and it is not essential to store all historical data.

Replay continual learning is adopted to alleviate the catastrophic forgetting of a single model for successive modes [17], [19], [26], where data in raw format or pseudo-samples from a generative model are replayed when a new mode arrives and the model needs to be retrained. In this paper, to decrease the storage and computational costs, a small amount of original data are selected from the past modes $\mathcal{M}_1, \cdots, \mathcal{M}_{K-1}$, which are sufficient to reflect the information of each mode and are replayed for future learning. For sequential modes, a single model is built based on data from existing modes, including the representative data from previous modes and normal data from the current mode $\mathcal{M}_K$. Since dynamic latent features are extracted from data in raw format, MNSDiPCA is able to monitor multiple diverse modes accurately with acceptable storage and computational costs.

III. Methodology

A. Selection of representative data

The key idea of the proposed approach is to select previous data for retraining so model training data sets are interleaved with current mode and selected data from previous modes, to achieve continual learning ability of model. In order to achieve minimal/important data sets from previous modes, cosine similarity is adopted for data selection. If the cosine similarity S_{cos} is higher than α_{sim}, we regard that data have similar information. For arbitrary samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, cosine similarity is calculated by

$$S_{cos} = \frac{\left|\boldsymbol{x}_i^T \boldsymbol{x}_j\right|}{\|\boldsymbol{x}_i\| \, \|\boldsymbol{x}_j\|} \tag{8}$$

The procedure of selecting data is summarized as follows:

a) Randomly select a sample $\boldsymbol{x}_i$ from $\boldsymbol{X}$, put it into $\tilde{\boldsymbol{X}}$;
b) Randomly select a sample $\boldsymbol{x}_j$ from $\boldsymbol{X}$, calculate the cosine similarity (8) between $\boldsymbol{x}_j$ and every sample in $\tilde{\boldsymbol{X}}$. If the similarity is lower than α_{sim}, put it into $\tilde{\boldsymbol{X}}$, otherwise, discard it;
c) Repeat step b) until $\boldsymbol{X}$ is null.

B. Proposed MNSDiPCA for multiple modes

When the Kth mode arrives ($K = 2, \cdots$), collect normal data $\boldsymbol{X}_K^0$. Here, for the past $K-1$ modes, a small amount of original data $\tilde{\boldsymbol{X}}_1^0, \cdots, \tilde{\boldsymbol{X}}_{K-1}^0$ have been selected by Section III-A. $\boldsymbol{X}_K^0$ and the selected data are normalized to zero means and unit variances, which are denoted as $\boldsymbol{X}_K$ and $\tilde{\boldsymbol{X}}_1, \cdots, \tilde{\boldsymbol{X}}_{K-1}$ respectively. Construct $\boldsymbol{X}^K = \left\{\tilde{\boldsymbol{X}}_1, \cdots, \tilde{\boldsymbol{X}}_{K-1}, \boldsymbol{X}_K\right\} \in R^{N^K \times m}$ and N^K is the number of training samples. MNSDiPCA aims to build a single monitoring model on $\boldsymbol{X}^K$ to monitor K modes simultaneously.

Data $\boldsymbol{X}^K$ are mapped to high-dimensional feature space by using (1), which are denoted as $\boldsymbol{X}_{\phi,K} \in R^{N^K \times M}$. Similar to (5) and (6), construct $\boldsymbol{X}_{\phi,K}^{(j)}$ ($j = 1, \cdots, s+1$) and $\boldsymbol{Z}_K = \left[\boldsymbol{X}_{\phi,K}^{(1)} \ \ \boldsymbol{X}_{\phi,K}^{(2)} \ \cdots \ \boldsymbol{X}_{\phi,K}^{(s)}\right]$. For K modes, the objective function of MNSDiPCA is designed as

$$\begin{aligned}&J_K(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{X}_{\phi,K}) \\ &= -\boldsymbol{w}^T \left(\boldsymbol{X}_{\phi,K}^{(s+1)}\right)^T \boldsymbol{Z}_K (\boldsymbol{\beta} \otimes \boldsymbol{w}) + \lambda_1 \boldsymbol{\beta}^T \boldsymbol{D} \boldsymbol{\beta}\end{aligned} \tag{9}$$

with the constraint $\boldsymbol{w}^T \boldsymbol{w} = 1$, $\boldsymbol{\beta}^T \boldsymbol{\beta} = 1$. ADMM [27] in Section III-C is utilized to settle the optimization issue (9) and the parameters are iteratively updated.

Replay methods store raw data [26], [28] or generate pseudo-samples via a generative model [29], which are replayed to alleviate forgetting when a new mode appears. Within the framework of multivariate statistic analysis, it may be not appropriate to train another separate generative model. To reduce the storage space and computation, only limited data are selected based on similarity and are sufficient to reflect the operating condition of each mode. Then, data from different modes are normalized by their means and variances, which are then integrated and regarded as one mode. Since the processed data cover sufficient information from all existing modes, the single monitoring model could provide excellent performance for successive diverse modes.

In comparison, regularization-based methods introduce an extra regularization term in the loss function to make the parameters close to the previous ones and preserve significant features of previous modes. Therefore, they are effective for multiple modes when data from different modes have a certain degree of similarity [12], [21]. One key point is to accurately evaluate the importance of model parameters and select proper hyperparameters, which would influence the monitoring performance. However, if the modes are diverse, the previously learned knowledge fails to provide valid information for future new modes and the model needs to be retrained from scratch. The proposed MNSDiPCA method is free from this constraint due to algorithmic simplicity and replay mechanism for continual learning, which makes it appropriate for long-term monitoring tasks. In addition, the estimated parameters are less than regularization-based methods, and thus requiring less manual intervention.

C. Optimization procedure by ADMM

The parameters $\boldsymbol{w}$ and $\boldsymbol{\beta}$ are optimized alternatively [27]. The weighting matrix $\boldsymbol{D}$ is updated after each iteration. Assume that the parameters $\boldsymbol{w}^i$, $\boldsymbol{z}_w^i$, $\boldsymbol{u}_w^i$, $\boldsymbol{\beta}^i$, $\boldsymbol{z}_\beta^i$ and $\boldsymbol{u}_\beta^i$ are known after ith iteration, the updating procedure at $(i+1)$th iteration is described below.

1) Update parameters about $\boldsymbol{w}$

$$\begin{aligned}\arg\min_{\boldsymbol{w}} \quad & J_K(\boldsymbol{w}, \boldsymbol{\beta}^i, \boldsymbol{X}_{\phi,K}) \\ s.t. \quad & \boldsymbol{w}^T\boldsymbol{w} = 1\end{aligned} \tag{10}$$

According to chapter 9 in [27], ADMM has the form:

$$\begin{aligned}\boldsymbol{w}^{i+1} &:= \arg\min_{\boldsymbol{w}} \left(J_K(\boldsymbol{w}, \boldsymbol{\beta}^i, \boldsymbol{X}_{\phi,K}) + \rho_w \left\|\boldsymbol{w} - \boldsymbol{z}_w^i + \boldsymbol{u}_w^i\right\|_2^2 \right) \\ \boldsymbol{z}_w^{i+1} &:= \frac{\boldsymbol{w}^{i+1} + \boldsymbol{u}_w^i}{\|\boldsymbol{w}^{i+1} + \boldsymbol{u}_w^i\|_2} \\ \boldsymbol{u}_w^{i+1} &:= \boldsymbol{u}_w^i + \boldsymbol{w}^{i+1} - \boldsymbol{z}_w^{i+1}\end{aligned} \tag{11}$$

where

$$\begin{aligned}& J_K(\boldsymbol{w}, \boldsymbol{\beta}^i, \boldsymbol{X}_{\phi,K}) + \rho_w \left\|\boldsymbol{w} - \boldsymbol{z}_w^i + \boldsymbol{u}_w^i\right\|_2^2 \\ = & -\boldsymbol{w}^T \left(\boldsymbol{X}_{\phi,K}^{s+1}\right)^T \boldsymbol{Z}_K \left(\boldsymbol{\beta}^i \otimes \boldsymbol{w}\right) + \rho_w \left\|\boldsymbol{w} - \boldsymbol{z}_w^i + \boldsymbol{u}_w^i\right\|_2^2\end{aligned}$$

and ρ_w is a regularization coefficient and predefined by users. Taking the derivative with regard to $\boldsymbol{w}$ and let it be zero, then

$$\boldsymbol{w}^{i+1} = 2\rho_w \left(\boldsymbol{G}_{\beta,K} + \boldsymbol{G}_{\beta,K}^T - 2\rho_w \boldsymbol{I}_M\right)^{-1} \left(\boldsymbol{u}_w^i - \boldsymbol{z}_w^i\right) \tag{12}$$

where $\boldsymbol{G}_{\beta,K} = \sum_{j=1}^{s} (\boldsymbol{X}_{\phi,K}^{(s+1)})^T \boldsymbol{X}_{\phi,K}^{(j)} \beta_j$.

Algorithm 1 Solution of MNSDiPCA based on ADMM

Require: Data $\boldsymbol{X}_{\phi,K}$, l, s.

Ensure: Weight matrix $\boldsymbol{W}$, loading matrix $\boldsymbol{P}$ and latent score matrix $\boldsymbol{T}$.

1: Construct $\boldsymbol{X}_{\phi,K}^{(j)}$ $(j = 1, \cdots, s+1)$ and $\boldsymbol{Z}_K$ by (5) and (6), let $g = 1$.
2: Initialize $\boldsymbol{w}^0$ and $\boldsymbol{\beta}^0$ with unit vector, $\boldsymbol{z}_w^0 = \boldsymbol{w}^0$, $\boldsymbol{u}_w^0 = \boldsymbol{0}$, $\boldsymbol{z}_\beta^0 = \boldsymbol{\beta}^0$, $\boldsymbol{u}_\beta^0 = \boldsymbol{0}$, $i = 0$.
3: Extract the dynamic component one by one:
   a) Calculate $\boldsymbol{w}^{i+1}$, $\boldsymbol{z}_w^{i+1}$ and $\boldsymbol{u}_w^{i+1}$ by (11) and (12);
   b) Calculate $\boldsymbol{\beta}^{i+1}$, $\boldsymbol{z}_\beta^{i+1}$ and $\boldsymbol{u}_\beta^{i+1}$ by (14) and (15);
   c) Update the weighting matrix $\boldsymbol{D}$ by (16);
   d) Calculate the objective function (9). Let $i = i+1$, return to step 3a) until convergence.
4: The optimal parameters are denoted as $\boldsymbol{w}_g$ and $\boldsymbol{\beta}_g$, and calculate the loading vector $\boldsymbol{p}_g = \frac{\boldsymbol{X}_{\phi,K}^T \boldsymbol{X}_{\phi,K} \boldsymbol{w}_g}{\boldsymbol{w}_g^T \boldsymbol{X}_{\phi,K}^T \boldsymbol{X}_{\phi,K} \boldsymbol{w}_g}$.
5: Let $\boldsymbol{t}_g = \boldsymbol{X}_{\phi,K}\boldsymbol{w}_g$, deflate $\boldsymbol{X}_{\phi,K}$ as $\boldsymbol{X}_{\phi,K} = \boldsymbol{X}_{\phi,K} - \boldsymbol{X}_{\phi,K}\boldsymbol{w}_g\boldsymbol{p}_g^T$ and then $(\boldsymbol{X}_{\phi,K}^{(s+1)})^T \boldsymbol{X}_{\phi,K}^{(j)}$ is calculated by (17).
6: Let $g = g+1$, return to step 2 until extracting l dynamic components.
7: The parameters are denoted as $\boldsymbol{W} = [\boldsymbol{w}_1 \ \cdots \ \boldsymbol{w}_l]$, $\boldsymbol{P} = [\boldsymbol{p}_1 \ \cdots \ \boldsymbol{p}_l]$, and $\boldsymbol{T} = [\boldsymbol{t}_1 \ \cdots \ \boldsymbol{t}_l]^T$.

---

2) Update parameters about $\boldsymbol{\beta}$

$$\begin{aligned}\arg\min_{\boldsymbol{\beta}} \quad & J_K(\boldsymbol{w}^{i+1}, \boldsymbol{\beta}, \boldsymbol{X}_{\phi,K}) \\ s.t. \quad & \boldsymbol{\beta}^T\boldsymbol{\beta} = 1\end{aligned} \tag{13}$$

Here, ADMM has the form [27]:

$$\begin{aligned}\boldsymbol{\beta}^{i+1} &:= \arg\min_{\boldsymbol{\beta}} \left(J_K(\boldsymbol{w}^{i+1}, \boldsymbol{\beta}, \boldsymbol{X}_{\phi,K}) + \rho_\beta \left\|\boldsymbol{\beta} - \boldsymbol{z}_\beta^i + \boldsymbol{u}_\beta^i\right\|_2^2 \right) \\ \boldsymbol{z}_\beta^{i+1} &:= \frac{\boldsymbol{\beta}^{i+1} + \boldsymbol{u}_\beta^i}{\left\|\boldsymbol{\beta}^{i+1} + \boldsymbol{u}_\beta^i\right\|_2} \\ \boldsymbol{u}_\beta^{i+1} &:= \boldsymbol{u}_\beta^i + \boldsymbol{\beta}^{i+1} - \boldsymbol{z}_\beta^{i+1}\end{aligned} \tag{14}$$

where

$$\begin{aligned}& J_K(\boldsymbol{w}^{i+1}, \boldsymbol{\beta}, \boldsymbol{X}_{\phi,K}) + \rho_\beta \left\|\boldsymbol{\beta} - \boldsymbol{z}_\beta^i + \boldsymbol{u}_\beta^i\right\|_2^2 \\ = & -(\boldsymbol{w}^{i+1})^T \boldsymbol{G}_K \left(\boldsymbol{\beta} \otimes \boldsymbol{w}^{i+1}\right) + \lambda_1 \boldsymbol{\beta}^T \boldsymbol{D}^i \boldsymbol{\beta} \\ & + \rho_\beta \left\|\boldsymbol{\beta} - \boldsymbol{z}_\beta^i + \boldsymbol{u}_\beta^i\right\|_2^2\end{aligned}$$

where ρ_β is a regularization parameter. Taking the derivative with regard to $\boldsymbol{\beta}$ and let it be zero, then

$$\begin{aligned}\boldsymbol{\beta}^{i+1} = & \left(\lambda_1 \boldsymbol{D}^i + \rho_\beta \boldsymbol{I}_s\right)^{-1} \\ & \left(\frac{1}{2}\left(\boldsymbol{I}_s \otimes \boldsymbol{w}^{i+1}\right)^T \boldsymbol{G}_K^T \boldsymbol{w}^{i+1} + \rho_\beta \left(\boldsymbol{z}_\beta^i - \boldsymbol{u}_\beta^i\right)\right)\end{aligned} \tag{15}$$

where $\boldsymbol{G}_K = \left(\boldsymbol{X}_{\phi,K}^{(s+1)}\right)^T \boldsymbol{Z}_K$ and $\left(\boldsymbol{I}_s \otimes \boldsymbol{w}^{i+1}\right)^T \boldsymbol{G}_K^T = \left[\left(\boldsymbol{X}_{\phi,K}^{(s+1)}\right)^T \boldsymbol{X}_{\phi,K}^{(1)} \boldsymbol{w}^{i+1} \ \cdots \ \left(\boldsymbol{X}_{\phi,K}^{(s+1)}\right)^T \boldsymbol{X}_{\phi,K}^{(s)} \boldsymbol{w}^{i+1}\right]^T$.

3) Update $\boldsymbol{D}$

$$\begin{aligned}\boldsymbol{D}^{i+1} &= diag\left\{d_1^{i+1}, d_2^{i+1}, \cdots, d_s^{i+1}\right\} \\ d_j^{i+1} &= \frac{1}{\left|\beta_j^{i+1}\right| + \epsilon}, \quad j = 1, \cdots, s\end{aligned} \tag{16}$$

Algorithm 2 Off-line training procedure of MNSDiPCA

1: When the Kth mode arrives, collect normal data $\boldsymbol{X}_K^0$. Calculate the means $\tilde{\boldsymbol{\mu}}_1, \cdots, \tilde{\boldsymbol{\mu}}_{K-1}, \boldsymbol{\mu}_K$ and variances $\tilde{\boldsymbol{\Sigma}}_1, \cdots, \tilde{\boldsymbol{\Sigma}}_{K-1}, \boldsymbol{\Sigma}_K$ of $\tilde{\boldsymbol{X}}_1^0, \cdots, \tilde{\boldsymbol{X}}_{K-1}^0, \boldsymbol{X}_K^0$. Data from each mode are normalized and denoted as $\tilde{\boldsymbol{X}}_1, \cdots, \tilde{\boldsymbol{X}}_{K-1}, \boldsymbol{X}_K$.
2: Construct $\boldsymbol{X}^K = \left\{\tilde{\boldsymbol{X}}_1, \cdots, \tilde{\boldsymbol{X}}_{K-1}, \boldsymbol{X}_K\right\}$, map $\boldsymbol{X}^K$ to high-dimensional feature space and get $\boldsymbol{X}_{\phi,K}^0$ by using (1).
3: Calculate the mean $\boldsymbol{\mu}_K^{\phi}$ and variance $\boldsymbol{\Sigma}_K^{\phi}$ of $\boldsymbol{X}_{\phi,K}^0$, and the normalized data are denoted as $\boldsymbol{X}_{\phi,K}$.
4: Conduct Algorithm 1 to settle the optimization issue (9), and get the parameters $\boldsymbol{W}$, $\boldsymbol{P}$, and $\boldsymbol{T}$.
5: Build a VAR model for latent variables and $\boldsymbol{T}_{s+1}$ is predicted by (20).
6: Perform PCA on prediction error matrix $\boldsymbol{E}$ in (24).
7: Calculate two test statistics by (22) and (25).
8: Calculate the corresponding thresholds by KDE.
9: Select the representative data $\tilde{\boldsymbol{X}}_K^0$ from $\boldsymbol{X}_K^0$ according to Section III-A.

where ϵ is a small positive value for numerical stability.

The solution of MNSDiPCA is summarized in Algorithm 1, where the dynamic components are acquired one by one. When $K = 1$, let $\boldsymbol{X}^1 = \boldsymbol{X}_1$ and the aforementioned solution is also applied. Note that we only need to map the original data into a high-dimensional feature space once. When $g \geq 2$, $\boldsymbol{X}_{\phi,K}$ is reconstructed by $\boldsymbol{X}_{\phi,K} - \boldsymbol{X}_{\phi,K}\boldsymbol{w}_g\boldsymbol{p}_g^T$. Once a dynamic component is extracted, $(\boldsymbol{X}_{\phi,K}^{(s+1)})^T\boldsymbol{X}_{\phi,K}^{(j)}$ $(j = 1, \cdots, s)$ is calculated recursively by

$$\begin{aligned}&(\boldsymbol{X}_{\phi,K}^{(s+1)})^T\boldsymbol{X}_{\phi,K}^{(j)}\\=&(\boldsymbol{X}_{\phi,K}^{(s+1)})^T\boldsymbol{X}_{\phi,K}^{(j)} - \boldsymbol{p}_g^T\boldsymbol{w}_g^T(\boldsymbol{X}_{\phi,K}^{(s+1)})^T\boldsymbol{X}_{\phi,K}^{(j)}\\&-(\boldsymbol{X}_{\phi,K}^{(s+1)})^T\boldsymbol{X}_{\phi,K}^{(j)}\boldsymbol{w}_g\boldsymbol{p}_g + \boldsymbol{p}_g^T\boldsymbol{w}_g^T(\boldsymbol{X}_{\phi,K}^{(s+1)})^T\boldsymbol{X}_{\phi,K}^{(j)}\boldsymbol{w}_g\boldsymbol{p}_g\end{aligned} \tag{17}$$

instead of recursion of $\boldsymbol{X}^K$ and mapping data repeatedly.

D. MNSDiPCA for process monitoring

Similar to DiPCA [5], define the latent score $\boldsymbol{t} = \boldsymbol{X}_{\phi,K}\boldsymbol{w}$ and matrix $\boldsymbol{T} = [\boldsymbol{t}_1 \ \cdots \ \boldsymbol{t}_l]^T$. Similar to (5), construct $\boldsymbol{T}_j$ from $\boldsymbol{T}$, $j = 1, \cdots, s+1$. Then, the dynamic relations between $\boldsymbol{T}_{s+1}$ and $\boldsymbol{T}_1, \cdots, \boldsymbol{T}_s$ can be represented by a VAR model, namely,

$$\begin{aligned}\boldsymbol{T}_{s+1} =&\boldsymbol{T}_1\boldsymbol{\Theta}_s + \boldsymbol{T}_2\boldsymbol{\Theta}_{s-1} + \cdots + \boldsymbol{T}_s\boldsymbol{\Theta}_1 + \boldsymbol{V}\\=&\bar{\boldsymbol{T}}_{\boldsymbol{s}}\boldsymbol{\Theta} + \boldsymbol{V}\end{aligned} \tag{18}$$

Algorithm 3 Online monitoring procedure of MNSDiPCA

1: Preprocess the testing sample $\boldsymbol{x}^0$ according to the mean value and variance of training data, which is then denoted as $\boldsymbol{x}$.
2: Map $\boldsymbol{x}$ to a high-dimensional space by (1) and get $\boldsymbol{x}_{\phi}$, which is preprocessed by $\boldsymbol{\mu}_K^{\phi}$ and $\boldsymbol{\Sigma}_K^{\phi}$.
3: Calculate the dynamic latent variable by (18), the prediction by (20) and the dynamic residual by (21).
4: Calculate test statistics by (22) and (25).
5: Confirm the operating condition: both statistics are below the thresholds, the process is normal; otherwise, faulty.

where $\bar{\boldsymbol{T}}_{\boldsymbol{s}} = [\boldsymbol{T}_1 \ \boldsymbol{T}_2 \ \cdots \ \boldsymbol{T}_s]$ and $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_s \ \boldsymbol{\Theta}_{s-1} \ \cdots \ \boldsymbol{\Theta}_1]$. The least squares estimate for $\boldsymbol{\Theta}$ is

$$\hat{\boldsymbol{\Theta}} = \left(\bar{\boldsymbol{T}}_{\boldsymbol{s}}{}^T\bar{\boldsymbol{T}}_{\boldsymbol{s}}\right)^{-1}\bar{\boldsymbol{T}}_{\boldsymbol{s}}{}^T\bar{\boldsymbol{T}}_{s+1} \tag{19}$$

Then, the prediction of $\boldsymbol{T}_{s+1}$ is calculated by

$$\hat{\boldsymbol{T}}_{s+1} = \bar{\boldsymbol{T}}_{\boldsymbol{s}}\hat{\boldsymbol{\Theta}} \tag{20}$$

Define the dynamic residual $\boldsymbol{V}$:

$$\boldsymbol{V} = \boldsymbol{T} - \hat{\boldsymbol{T}}_{s+1} \tag{21}$$

Perform PCA on $\boldsymbol{V}$ and obtain the principal component matrix $\boldsymbol{P}_v$, 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 statistics T_v^2 and $\mathrm{SPE_v}$, respectively. Similar to [12], an index is designed through $\boldsymbol{V}$ and calculated by

$$T_{\varphi}^2 = \boldsymbol{v}^T\boldsymbol{\Phi}_v\boldsymbol{v} \tag{22}$$

$$\boldsymbol{\Phi}_v = \frac{\boldsymbol{P}_v\boldsymbol{\Lambda}_v^{-1}\boldsymbol{P}_v^T}{J_{th,T_v^2}} + \frac{\boldsymbol{I} - \boldsymbol{P}_v\boldsymbol{P}_v^T}{J_{th,\mathrm{SPE_v}}} \tag{23}$$

The static prediction error is calculated by

$$\boldsymbol{E} = \boldsymbol{X}_{\phi,K}^{(s+1)} - \boldsymbol{T}_{s+1}\boldsymbol{P}^T \tag{24}$$

Similarly, perform PCA on $\boldsymbol{E}$, and get the principal component matrix $\boldsymbol{P}_r$ 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 $\mathrm{SPE_r}$. A monitoring index is calculated to monitor the static error accordingly.

$$T_c^2 = \boldsymbol{e}^T\boldsymbol{\Phi}_c\boldsymbol{e} \tag{25}$$

$$\boldsymbol{\Phi}_c = \frac{\boldsymbol{P}_r\boldsymbol{\Lambda}_r^{-1}\boldsymbol{P}_r^T}{J_{th,T_r^2}} + \frac{\boldsymbol{I} - \boldsymbol{P}_r\boldsymbol{P}_r^T}{J_{th,\mathrm{SPE_r}}} \tag{26}$$

Two thresholds are determined by kernel density estimation (KDE) [10], [12]. The training and online monitoring phases are summarized in Algorithms 2 and 3, respectively. The monitoring performance is evaluated by fault detection rate (FDR) and false alarm rate (FAR) in this paper.

IV. COMPARATIVE SCHEMES AND CASE STUDIES

A. Comparative analysis and experiment design

In this paper, take three sequential modes as an instance to manifest the effectiveness of the proposed method. SDiPCA–MSI [12], PCA–EWC [21], IMPPCA [10] and MCVA [11] are adopted as comparison. Similar to [30], the virtues and drawbacks of five methods are summarized in Table I. The performance is evaluated by the monitoring performance on new and previous modes, as designed in Table II.

Similar to [12], [21], Situations 1–11 are designed to illustrate the continual learning ability of MNSDiPCA and the catastrophic forgetting issue of NSDiPCA. For each mode, typical data are selected by cosine similarity and stored once the learning procedure finishes, which can reflect the operating condition. When a new mode is encountered, representative data are replayed and employed to build a single model, thus it needs moderate storage and computing resources. For instance, when the second mode $\mathcal{M}_2$ arrives, data $\boldsymbol{X}_2$ are collected and the stored data $\tilde{\boldsymbol{X}}_1$ are replayed for retraining to establish the model $\mathcal{B}$, which aims to monitor modes

TABLE II
COMPARATIVE SCHEMES AND SIMULATION RESULTS OF FOUR CASES (FDRS (%) AND FARS (%))

	Methods	Training sources (Model + data)	Model label	Testing sources	Case 1 FDR	Case 1 FAR	Case 2 FDR	Case 2 FAR	Case 3 FDR	Case 3 FAR	Case 4 FDR	Case 4 FAR
Situation 1	NSDiPCA	$\boldsymbol{X}_1$	$\mathcal{A}$	$\mathcal{M}_1$	99.80	4.20	100	1.60	100	0.75	100	1.27
Situation 2	MNSDiPCA	$\tilde{\boldsymbol{X}}_1, \boldsymbol{X}_2$	$\mathcal{B}$	$\mathcal{M}_2$	97.58	1.20	100	0.38	98.30	0	100	1.12
Situation 3	MNSDiPCA	-	$\mathcal{B}$	$\mathcal{M}_1$	100	3.80	100	4.48	100	1.61	100	6.09
Situation 4	NSDiPCA	$\boldsymbol{X}_2$	$\mathcal{C}$	$\mathcal{M}_2$	99.80	6.60	100	0.38	98.42	0.73	100	1.38
Situation 5	NSDiPCA	-	$\mathcal{C}$	$\mathcal{M}_1$	100	15.80	100	84.0	100	32.80	100	26.36
Situation 6	MNSDiPCA	$\tilde{\boldsymbol{X}}_1, \tilde{\boldsymbol{X}}_2, \boldsymbol{X}_3$	$\mathcal{D}$	$\mathcal{M}_3$	97.78	0.80	100	0.44	98.60	9.73	100	0.71
Situation 7	MNSDiPCA	-	$\mathcal{D}$	$\mathcal{M}_1$	100	2.40	100	8.16	100	1.12	100	0.45
Situation 8	MNSDiPCA	-	$\mathcal{D}$	$\mathcal{M}_2$	100	2.80	100	0.38	98.42	1.06	100	1.12
Situation 9	NSDiPCA	$\boldsymbol{X}_3$	$\mathcal{E}$	$\mathcal{M}_3$	100	4.40	100	4.52	99.24	24.40	100	47.69
Situation 10	NSDiPCA	-	$\mathcal{E}$	$\mathcal{M}_1$	100	15.40	100	73.6	100	19.88	100	62.82
Situation 11	NSDiPCA	-	$\mathcal{E}$	$\mathcal{M}_2$	100	14.40	100	0.38	98.48	13.02	100	8.87
Situation 12	SDiPCA	$\boldsymbol{X}_1$	$\mathcal{F}$	$\mathcal{M}_1$	100	2.20	100	0.80	100	10.56	100	0.01
Situation 13	SDiPCA–MSI	$\mathcal{F}$ + $\boldsymbol{X}_2$	$\mathcal{G}$	$\mathcal{M}_2$	100	0.40	100	0	98.30	1.63	100	0.25
Situation 14	SDiPCA–MSI	-	$\mathcal{G}$	$\mathcal{M}_1$	99.80	0.40	100	44.48	100	6.96	100	0.82
Situation 15	SDiPCA–MSI	$\mathcal{G}$ + $\boldsymbol{X}_3$	$\mathcal{H}$	$\mathcal{M}_3$	99.20	2.20	100	0.33	99.11	13.99	100	34.20
Situation 16	SDiPCA–MSI	-	$\mathcal{H}$	$\mathcal{M}_1$	99.00	2.40	100	28.16	100	5.71	100	16.64
Situation 17	SDiPCA–MSI	-	$\mathcal{H}$	$\mathcal{M}_2$	99.20	3.80	99.45	0	98.36	0.33	100	1.25
Situation 18	PCA	$\boldsymbol{X}_1$	$\mathcal{I}$	$\mathcal{M}_1$	92.80	7.80	100	1.44	100	0	99.71	0
Situation 19	PCA–EWC	$\mathcal{I}$ + $\boldsymbol{X}_2$	$\mathcal{J}$	$\mathcal{M}_2$	91.00	8.00	99.45	0	98.06	0	100	0.13
Situation 20	PCA–EWC	-	$\mathcal{J}$	$\mathcal{M}_1$	88.20	6.60	100	11.52	100	1.24	99.71	2.18
Situation 21	PCA–EWC	$\mathcal{J}$ + $\boldsymbol{X}_3$	$\mathcal{L}$	$\mathcal{M}_3$	93.20	7.20	99.95	2.75	98.92	10.75	100	37.51
Situation 22	PCA–EWC	-	$\mathcal{L}$	$\mathcal{M}_1$	93.40	7.80	100	38.88	100	0.99	99.70	16.73
Situation 23	PCA–EWC	-	$\mathcal{L}$	$\mathcal{M}_2$	94.60	9.40	99.45	0	98.12	2.79	100	1.63
Situation 24	IMPPCA	$\boldsymbol{X}_1, \boldsymbol{X}_2$	$\mathcal{N}$	$\mathcal{M}_1$	67.40	3.80	100	37.44	100	3.48	99.71	0.36
Situation 25	IMPPCA	-	$\mathcal{N}$	$\mathcal{M}_2$	54.80	4.20	99.45	0	98.18	1.71	100	1.63
Situation 26	IMPPCA	$\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$	$\mathcal{O}$	$\mathcal{M}_1$	98.40	1.40	100	10.40	100	2.36	99.71	0.45
Situation 27	IMPPCA	-	$\mathcal{O}$	$\mathcal{M}_2$	100	13.00	99.45	0	98.24	7.73	100	0.38
Situation 28	IMPPCA	-	$\mathcal{O}$	$\mathcal{M}_3$	96.00	0	100	1.10	98.73	25.26	100	69.23
Situation 29	MCVA	$\boldsymbol{X}_1, \boldsymbol{X}_2$	$\mathcal{P}$	$\mathcal{M}_1$	100	15.20	100	10.72	100	23.98	99.71	5.82
Situation 30	MCVA	-	$\mathcal{P}$	$\mathcal{M}_2$	58.27	14.80	100	4.14	98.12	17.01	100	0.50
Situation 31	MCVA	$\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$	$\mathcal{Q}$	$\mathcal{M}_1$	100	14.60	100	7.84	100	23.98	99.71	7.82
Situation 32	MCVA	-	$\mathcal{Q}$	$\mathcal{M}_2$	57.06	14.20	100	3.95	98.12	20.83	100	6.00
Situation 33	MCVA	-	$\mathcal{Q}$	$\mathcal{M}_3$	24.19	12.20	99.95	28.30	99.94	34.13	100	62.72

$\mathcal{M}_1$ and $\mathcal{M}_2$ simultaneously. Thus, the continual learning ability of MNSDiPCA is reflected. Correspondingly, Situation 5 is designed to illustrate the catastrophic forgetting issue of NSDiPCA, namely, the model $\mathcal{C}$ built for mode $\mathcal{M}_2$ should fail to monitor the mode $\mathcal{M}_1$. Situations 6–11 are designed in a similar manner. Since significant features are assimilated from data in raw format that are collected from the existing modes, it can provide excellent performance for new and previous modes simultaneously based on a single model.

For Situations 12–23, SDiPCA–MSI and PCA–EWC provide continual learning ability based on a continually updated model for multimode linear processes. A regularization term is added to prevent the mode-sensitive parameters from changing dramatically, thus avoiding a sudden performance degradation for the previously learned modes. SDiPCA–MSI and PCA–EWC discard raw data after finishing the learning procedure and only store the learned knowledge. Therefore, they need the least storage and computing resources among the comparative methods. However, they require similarity among various modes and need to retrain the model from scratch once the modes are especially diverse [15], which makes them appropriate for short-term monitoring tasks.

IMPPCA and MCVA require complete data from all potential modes before learning and the model may need to be retrained from scratch when a new mode arrives. Since all observation data are stored and would be utilized for retraining in future, they need expensive storage space and computing resources with the successive emergence of modes. Different from three aforementioned methods, a global monitoring model is built based on a weighted sum of local models and the weight could be evaluated by posterior probability.

TABLE I
COMPARISON OF FIVE MULTIMODE MONITORING METHODS

	MNSDiPCA	SDiPCA–MSI PCA–EWC	IMPPCA MCVA
new mode performance	good	good	good
previous mode performance	good	medium	good
training efficiency	fast	fast	slow
testing efficiency	fast	fast	fast
storage space	medium	low	large
require previous mode data	yes	no	yes

B. CSTH case

The CSTH process is a popular benchmark that is widely utilized for monitoring multimode dynamic processes. It mixed the cold water and hot water together to obtain the desired product. Three key parameters are controlled by PI controllers, namely, level, temperature and flow. Detailed information can refer to [31]. In this paper, three sequential modes are designed in Table III. A step fault occurs in the level from the $501th$ sample and the fault amplitude is 0.05.

TABLE III
NORMAL OPERATING MODES OF CSTH

Case number	Mode label	Level SP	Temperature SP	Hot water valve
Case 1	$\mathcal{M}_1$	11	10.5	5
	$\mathcal{M}_2$	13	10.5	4
	$\mathcal{M}_3$	12	11	5

The detail settings of five methods are listed in the supplementary material owing to the limitation of paper length and the monitoring results are listed in Table II. According to Situations 1–11, MNSDiPCA is capable of monitoring multiple modes accurately based on a single model, where the FDRs are higher than 97.5% and the FARs are lower than 4%. In addition, the FARs of Situations 4 and 8 are 6.60% and 2.80%, which indicates that the information of other modes ($\mathcal{M}_1$ and $\mathcal{M}_3$) is beneficial to enhancing the detection accuracy of mode $\mathcal{M}_2$. However, NSDiPCA suffers from the catastrophic forgetting issue in multimode process, where the FARs of Situations 5, 10 and 11 are higher than 14%. Briefly speaking, the NSDiPCA model based on one mode data fails to detect the faults in another mode, and the previously learned knowledge would be overwritten. SDiPCA–MSI is able to monitor this case, and the FDRs approach 100% and the FARs are less than 4%. PCA–EWC cannot offer excellent performance, where the FDRs are lower than 95% and the FARs are higher than 6.50%. IMPPCA and MCVA are not able to monitor this case, and the FDRs of Situations 24, 25, 30, 32 and 33 are lower than 70%. The monitoring charts of Case 1 are described in Figure 1 in the supplementary material owing to the limitation of paper length.

Different from four comparative methods, MNSDiPCA needs to select representative data from each mode and the number of selected data is listed in Table IV. Let $\alpha_{sim} = 0.9$, less than 8% of the training samples are stored and would be replayed for future learning, which allows it to cost limited computing and storage resources. SDiPCA–MSI and PCA–EWC would discard training data once the learning procedure finishes. Conversely, IMPPCA and MCVA need to store all training data and would be used when a novel mode arrives, thus they require the most expensive computing and storage costs. For Case 1, the proposed MNSDiPCA and SDiPCA–MSI may be optimal for successive dynamic processes, in terms of detection accuracy, computing and storage costs.

TABLE IV
STORED DATA OF PREVIOUS MODES BASED ON COSINE SIMILARITY

Data	$\tilde{\boldsymbol{X}}_1$ ($\mathcal{M}_1$)	$\tilde{\boldsymbol{X}}_2$ ($\mathcal{M}_2$)	$\tilde{\boldsymbol{X}}_3$ ($\mathcal{M}_3$)
Case 1	79	76	74
Case 2	318	54	108
Case 3	247	171	225
Case 4	128	122	34

C. Coal pulverizing system

This paper utilizes real data from the coal pulverizing systems to illustrate the effectiveness of the proposed method, which is one crucial unit of the 1030–MW ultra-supercritical thermal power plant and locates at Zhoushan, Zhejiang Province, China. The schematic diagram of the system has been depicted in Figure 2 in the supplementary material owing to the paper length, which is composed of coal feeder, coal mill, rotary separator, raw coal hopper and stone coal scuttle. To improve combustion efficiency and guarantee operating safety, the raw coal is ground into pulverized coal with desired temperature and fineness. In accordance with the historical records and demands, three types of faults are considered, including novelty from outlet temperature (Case 2), rotary separator (Case 3) and the coal feeder (Case 4). The data information is outlined in Table V and the key variables are selected by expert experience.

The experiment design and the monitoring results of three cases are summarized in Table II. The detail settings of five methods are listed in the supplementary material owing to the limitation of paper length. Take Case 4 as an example to illustrate the results specifically and partial monitoring charts are described in Fig. 1. When the mode $\mathcal{M}_2$ arrives, MNSDiPCA establishes a single model based on a few representative data $\tilde{\boldsymbol{X}}_1$ and data $\boldsymbol{X}_2$, which provides excellent performance for two modes simultaneously. The FDRs of Situations 2 and 3 are 100%. However, the NSDiPCA model based on data $\boldsymbol{X}_2$ is not able to monitor the previous mode $\mathcal{M}_1$, and the FAR of Situation 5 is 26.36%. SDiPCA–MSI, PCA–EWC, IMPPCA and MCVA can monitor two modes $\mathcal{M}_1$ and $\mathcal{M}_2$ accurately. The FDRs of Situations 13, 14, 19, 20, 24, 25, 29 and 30 are perfect. Besides, the FARs are lower than 6%. When the mode $\mathcal{M}_3$ appears, the MNSDiPCA model is retrained from scratch and enables to detect the faults in three modes. The FDRs of Situations 6–8 are 100% and the FARs are lower than 1.2%. However, the FAR of Situation 9 is 47.69%, which implies that it is hard to build an efficient NSDiPCA model $\mathcal{E}$ only based on normal data $\boldsymbol{X}_3$. Similarly, the model $\mathcal{E}$ fails to monitor the mode $\mathcal{M}_1$, which reflects the catastrophic forgetting issue of NSDiPCA. The performance of the remaining four methods is not acceptable, where the FARs of Situations 21, 28 and 33 are higher than 35%. The aforementioned analysis is also applied to Cases 2 and 3. Owing to the limitation of paper length, the partial monitoring charts of Cases 2 and 3 are described Figure 3 and Figure 4 in the supplementary material.

For the proposed MNSDiPCA, the representative data from each mode are selected by cosine similarity, as summarized in Table IV. Except mode $\mathcal{M}_1$ in Case 2, less than 10% of the samples are selected, which indicates that MNSDiPCA needs moderate storage resource. With regard to SDiPCA–MSI and PCA–EWC, there is no need to allocate additional space to store historical data, thus requiring the least storage resources. With the successive emergence of modes, the computational complexity of IMPPCA and MCVA is the highest, because the amount of training samples increases significantly and

TABLE V
EXPERIMENTAL DATA OF THE PRACTICAL COAL PULVERIZING SYSTEM

Case number	Key variables	Mode number	Number of training data	Number of testing data	Fault location	Coal type	Fault cause
Case 2	9 variables: pressure of air powder mixture, outlet temperature, primary air pressure and temperature, etc.	$\mathcal{M}_1$	1440	1440	626	Aomeng	Air leakage at primary air interface
		$\mathcal{M}_2$	1080	1080	533	Yinni	Hot primary air electric damper failure
		$\mathcal{M}_3$	2160	2880	909	Yinni	Pulverizer deflagration
Case 3	9 variables: rotary separator speed and current, coal feeding capacity, bearing temperature, etc.	$\mathcal{M}_1$	2880	1080	806	Aomeng	Frequency conversion cabinet short circuit
		$\mathcal{M}_2$	2880	2880	1230	Aomeng	Cooling fan trip of inverter cabinet
		$\mathcal{M}_3$	2880	2160	587	Shenhun	Frequency conversion cabinet short circuit
Case 4	14 variables: speed of coal feeder, rotary separator speed and current, coal feeding capacity, etc.	$\mathcal{M}_1$	2160	1440	1101	Shenhun	Coal block of the coal pipe
		$\mathcal{M}_2$	2520	1440	801	Aomeng	The coal feeder belt is broken
		$\mathcal{M}_3$	1080	1080	846	Aomeng	The coal feeder does not drop coal

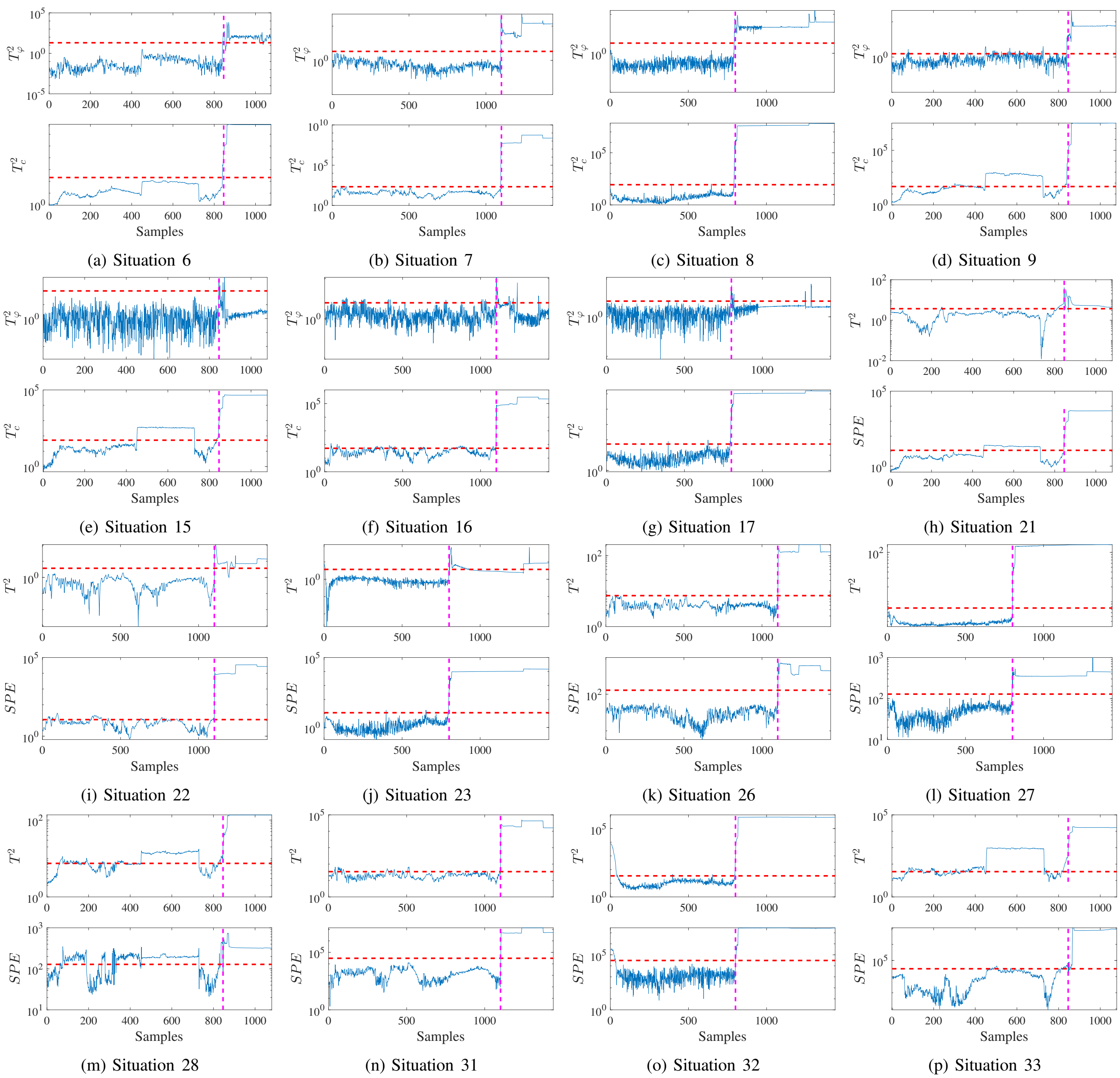

Fig. 1. Monitoring charts of Case 4

the model needs to be retrained from scratch when a new mode appears. Conversely, the computational complexity of SDiPCA–MSI and PCA–EWC is only related to the current mode data, and the complexity may be the least. The complexity of MNSDiPCA is middle because the number of training data may be much less than that of IMPPCA and MCVA.

In conclusion, MNSDiPCA, SDiPCA–MSI and PCA–EWC build a single model and provide continual learning ability for all existing modes. The major difference is the manner of preserving the significant information from previous modes. MNSDiPCA inherits the core of replay continual leaning [26], and only a few representative data are stored and utilized when a new mode arrives, which contain sufficient information of each mode. However, SDiPCA–MSI and PCA–EWC adopt the spirit of the regularization continual learning and build a continually updated model, where the parameters sensitive to modes are desired to change little. They require that data among different modes are similar to a certain degree. Once the modes are diverse, the performance would be decreased abruptly [15]. MNSDiPCA is free from this constraint, and thus may be better than SDiPCA–MSI and PCA–EWC for Cases 2–4. IMPPCA and MCVA need complete data from all modes before learning and the mode may be misidentified when the modes are relatively similar.

V. Conclusion

This paper has introduced an efficient nonlinear dynamic method with continual learning ability for multimode process monitoring, where data from different modes are collected in a sequential fashion. A few data from previous modes have been selected by cosine similarity and would be replayed together with the current mode data to extract the significant features when a new mode is encountered. Therefore, it is able to deliver outstanding performance for the existing diverse modes based on a single model. A pre-learning step is conducted to map the original data into a high-dimensional feature space to settle the nonlinearity. In addition, a regularization term is added to avoid the potential overfitting issue and the catastrophic forgetting issue may be further alleviated simultaneously. In comparison to traditional multimode methods, the proposed approach can alleviate the storage burden and reduce the computational cost. Compared with several state-of-the-art multimode monitoring methods, the effectiveness of the proposed MNSDiPCA algorithm is demonstrated via CSTH case and a practical coal pulverizing system.

In this work it has been assumed that the mode information is available and sufficient data are collected before retraining the model. In our future work, the mode identification would be investigated automatically and an adaptive method with continual learning ability will be investigated for online real-time monitoring.

References

[1] M. Quiñones-Grueiro, O. Llanes-Santiago, and A. J. S. Neto, *Monitoring Multimode Continuous Processes: A Data-Driven Approach*. Springer, 2020.

[2] K. E. S. Pilario and Y. Cao, “Canonical variate dissimilarity analysis for process incipient fault detection,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5308–5315, 2018.

[3] Y. Jiang and S. Yin, “Recent advances in key-performance-indicator oriented prognosis and diagnosis with a MATLAB toolbox: DB-KIT,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2849–2858, 2019.

[4] D. Wu, D. Zhou, and M. Chen, “Probabilistic stationary subspace analysis for monitoring nonstationary industrial processes with uncertainty,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3114–3125, 2022.

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

[6] P.-E. Odiowei and Y. Cao, “Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations,” *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 36–45, 2010.

[7] G. Wang, J. Jiao, and S. Yin, “Efficient nonlinear fault diagnosis based on kernel sample equivalent replacement,” *IIEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2682–2690, 2019.

[8] Y. Liu, J. Zeng, J. Bao, and L. Xie, “A unified probabilistic monitoring framework for multimode processes based on probabilistic linear discriminant analysis,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6291–6300, 2020.

[9] 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 Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1989–2003, 2020.

[10] J. Zhang, H. Chen, S. Chen, and X. Hong, “An improved mixture of probabilistic PCA for nonlinear data-driven process monitoring,” *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 198–210, 2019.

[11] Q. Wen, Z. Ge, and Z. Song, “Multimode dynamic process monitoring based on mixture canonical variate analysis model,” *Ind. Eng. Chem. Res.*, vol. 54, no. 5, pp. 1605–1614, 2015.

[12] J. Zhang, D. Zhou, M. Chen, and X. Hong, “Continual learning for multimode dynamic process monitoring with applications to an ultra-supercritical thermal power plant,” *IEEE Trans. Autom. Sci. Eng.*, doi: 10.1109/TASE.2022.3144288, 2022.

[13] R. Tan, J. R. Ottewill, and N. F. Thornhill, “Nonstationary discrete convolution kernel for multimodal process monitoring,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3670–3681, 2020.

[14] B. Wang, Z. Li, Z. Dai, N. Lawrence, and X. Yan, “Data-driven mode identification and unsupervised fault detection for nonlinear multimode processes,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3651–3661, 2020.

[15] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, “Embracing change: Continual learning in deep neural networks,” *Trends Cogn. Sci*, vol. 24, no. 12, pp. 1028–1040, 2020.

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

[17] 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 Trans. Pattern Anal. Mach. Intell*, vol. 44, no. 7, pp. 3366–3385, 2022.

[18] 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,” *Proc. Natl. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

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

[20] N. Y. Masse, G. D. Grant, and D. J. Freedman, “Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization,” *Proc. Natl. Acad. Sci. USA*, vol. 115, no. 44, pp. E10 467–E10 475, 2018.

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

[22] J. Zhang, D. Zhou, and M. Chen, “Self-learning sparse PCA for multimode process monitoring,” *IEEE Trans. Ind. Informat.*, doi:10.1109/TII.2022.3178736, 2022.

[23] J. Zhang, D. Zhou, and M. Chen, “Adaptive cointegration analysis and modified RPCA with continual learning ability for monitoring multi-mode nonstationary processes,” *IEEE Trans. Cyber.*, doi: 10.1109/TCY-B.2021.3140065, 2022.

[24] F. L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 11, no. 6, pp. 567–585, 1989.

[25] L. Wiskott and T. J. Sejnowski, “Slow feature analysis: Unsupervised learning of invariances,” *Neural Comput.*, vol. 14, no. 4, pp. 715–770, 2002.

[26] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–20.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Foundations and Trends, 2011.

[28] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 350–360, 2019.

[29] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, 2017, pp. 2990–2999.

[30] Z. Li and D. Hoiem, "Learning without forgetting," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 614–629.

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

Jingxin Zhang received B.E. degree in School of Electrical Engineering and Automation from Harbin Engineering University, Harbin, China, the M.E. degree in Control Science and Engineering from Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively, and the Ph.D. degree in Control Science and Engineering from Tsinghua University, Beijing, China, in 2022. She is currently a lecture with the Department of Automation, Southeast University. Her research interests are conintual learning, data-driven fault detection and diagnosis, performance monitoring and their applications in the industrial process.

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.