A hierarchical multimode dynamic process monitoring scheme and its application to the Tennessee Eastman process ⋆

Jiaorao Wang *, **Lishuai Li** **, **S. Joe Qin** *

* *School of Data Science, Lingnan University, Hong Kong*
** *School of Data Science, City University of Hong Kong, Hong Kong*

Abstract: Multimode characteristics commonly exist in modern industrial processes. Previous multi-model approaches treat steady states and transitions separately. However, identifying each mode is often tedious, generally achieved through clustering, requiring operators to tune hyperparameters extensively. As practitioners prefer a concise and easily implemented approach for multimode dynamic process monitoring, we initially propose a hierarchical scheme to simplify the modeling process while enhancing monitoring performance. Our method iteratively constructs dynamic models in a hierarchical, monitoring-oriented manner without mode partition. It offers three advantages. Firstly, modeling is directly conducted following a hierarchical structure driven by monitoring indexes, which is more concise and ensures monitoring performance. Secondly, by eliminating mode partition, only three hyperparameters, such as model order and the termination condition, need to be decided by humans. This significantly reduces human labour and facilitates the applicability of the proposed method across various processes. Lastly, by focusing on dynamic characteristics rather than steady-state and transitional modes, our method reduces the number of required models for a given process, resulting in a simpler multi-model structure that still ensures monitoring performance.

Keywords: Multimode dynamic process monitoring, fault detection, dynamic modeling, hierarchical scheme, autoregressive models.

1. INTRODUCTION

In modern industrial applications, processes often exhibit multimode properties due to changes in operating conditions, production requirements, or equipment configurations. A multimode dynamic process operates by switching between distinct steady states, each representing a specific operational mode. The switching process is called transition. These multimode characteristics introduce significant complexity into process monitoring and control. Accurately monitoring and detecting faults across multiple modes are essential for ensuring safety, maintaining product quality, and improving overall efficiency in industrial systems.

Various methods have been developed to address the challenges of multimode process monitoring, broadly categorized into single-model and multi-model schemes (Quiñones-Grueiro et al., 2019). In the single-model scheme, adaptive modeling approaches using neural networks are widely applied (Wu and Zhao, 2020; Song et al., 2024). For example, Huang et al. (2023) utilized a jointly mode-matching and similarity-preserving dictionary learning to learn the data of new modes, while trying to guarantee the representation ability of the proposed method for historical data. Although adaptive approaches using neural networks can provide flexibility in adapting to changing conditions, they have three drawbacks. The first one is that neural networks lack interpretability. It is hard to understand the decisions. Secondly, adaptive methods based on neural networks are memory-intensive and struggle to catch process changes quickly. They inevitably have a lag in fault detection. Finally, for different industrial systems, neural networks often require retraining or fine-tuning, which can be a highly labour-intensive and time-consuming process. Therefore, practitioners prefer simple, structurally transparent models in practical applications.

To overcome the aforementioned limitations, the multiple-model scheme is a good choice and more popular among researchers. It separately treats steady states and transitions, which is clearer and more understandable than the single-model scheme. Typically, this scheme first partitions the multimode process into distinct modes and then develops a model, such as principal component analysis (PCA), independent component analysis (ICA) and canonical correlation analysis (CCA) model, for each mode (Lyu et al., 2024; Xu et al., 2023). Mode partition is essential to capture the unique characteristics of each operational mode. In cases where expert knowledge is unavailable, clustering and classification techniques are commonly employed to identify different modes (Wang et al., 2020; Chen et al., 2021). However, in these multi-model methods, both mode partition and modeling tend to overlook, to some extent, the dynamic behaviour inherent in process data, which can limit their effectiveness in distinguishing modes with

⋆ Corresponding author is S. Joe Qin (joeqin@ln.edu.hk). This work was supported partially by Innovation and Technology Commission (ITC) and partially by Guangdong-Hong Kong Technology Cooperation Funding Scheme (Project No. GHP/145/20).

different dynamic characteristics. Additionally, mode partition itself can be computationally demanding, requiring careful tuning and validation to achieve robust results.

In our previous work, dynamic characteristics of the multimode process have been emphasized (Wang et al., 2023). Local dynamic models are constructed for each data to represent corresponding dynamic patterns. Instead of treating steady-state and transitional modes respectively, we identify multiple dynamic models to represent different dynamic patterns involved in the multimode process. Although this approach significantly takes dynamic characteristics into account and reduces model numbers, i.e., simplifies the multi-model structure, it still needs to use clustering with many hyperparameters to differentiate dynamics, which is also complicated.

The approach that practitioners prefer to use for multimode process monitoring is the one with a concise model structure and easily implemented for different processes without prior knowledge. To satisfy this demand and overcome the shortcomings of complex training processes and numerous hyperparameters, we initially propose a hierarchical scheme for multimode dynamic process monitoring. The contributions of this work are threefold:

(1) The hierarchical algorithm directly constructs dynamic models for multimode process monitoring without data clustering and classification. Such a modeling process is monitoring-oriented and more concise than the existing multi-model approaches mentioned above.
(2) Only three hyperparameters including model order and the termination condition need human adjustment, which means the proposed method is convenient to employ in different processes.
(3) The proposed method treats different dynamic characteristics rather than steady states and transitions. It reduces the redundancy in the number of models while describing the process accurately. For the same process, our method performs better in fault detection with a simpler multi-model structure.

2. PRELIMINARIES AND PROBLEM STATEMENTS

To spell out exactly the improvement of the hierarchical scheme compared with existing multi-model approaches, we first review the modeling process of existing methods, and then describe our scheme along with problem statements.

2.1 Preliminaries

Let $X = \{x_1, \ldots, x_N\}$ represent N data points consisting of the training data set, where each $x_t \in \mathbb{R}^d$ is a d-dimensional vector. In the aforementioned multi-model algorithms, the first step is to classify X into S sets $X_1, \ldots, X_S$ according to extracted features Φ, and identify models $F_1, \ldots, F_S$ for different modes $M_1, \ldots, M_S$. Hence, the overall model can be denoted as

$$y_t = \begin{cases} F_1(x_t) & if \quad x_t \in M_1, \\ \vdots & \quad \vdots \\ F_S(x_t) & if \quad x_t \in M_S, \end{cases} \tag{1}$$

where y_i indicates the output of the overall model, and $x_t \in M_i$ means that data x_t belongs to mode i. We plot this modeling process in Fig. 1 to show it more clearly. This multi-model scheme in (1) includes two steps for modeling: One is the mode partition, usually achieved by clustering. Another is model identification. For clustering, extracting data features and distinguishing different features are tedious and varied across different processes, which obstructs the promotion of a method. Hence, we devise a new hierarchical scheme to simplify modeling process.

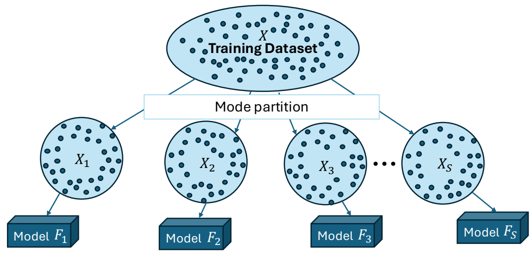


Fig. 1. Modeling process based on mode partition.

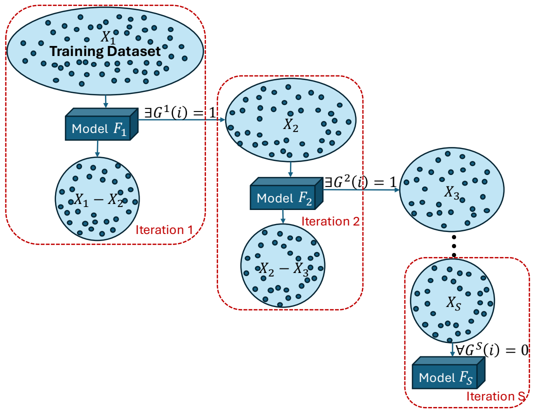


Fig. 2. Modeling process of the proposed hierarchical scheme.

The hierarchical scheme is a general framework described as follows. It is assumed that S iterations are conducted and the training dataset is initilalized as X_1. For each iteration j, a dynamic model F_j is identified based on the corresponding data set X_j. Then, a threshold condition is used to determine if another iteration is needed, which is defined as

$$I(t) = \begin{cases} 1 & if \quad \phi_t > \phi_{th}, \\ 0 & otherwise, \end{cases} \tag{2}$$

where ϕ_t is the monitoring index calculated for x_t and ϕ_{th} is the threshold. If there are n_c consecutive indices exceeding the threshold ϕ_{th}, then another iteration should be conducted. We define the consecutive exceedance counter as

$$C(t) = \begin{cases} C(t-1)+1 & if \quad I(t) = 1, \\ 0 & if \quad I(t) = 0. \end{cases} \tag{3}$$

Thus, an additional iteration is performed when $C(t) > n_c$ holds. Accordingly, the transition condition to the next level is defined as

$$G(t) = \begin{cases} 1 & if \quad C(t) > n_c, \\ 0 & otherwise. \end{cases} \tag{4}$$

Finally, the overall model obtained offline by the proposed hierarchical scheme can be denoted as

$$y_t = \begin{cases} F_1(x_t) & if \quad x_t \in X_1 - X_2, \\ \vdots & \quad\quad \vdots \\ F_S(x_t) & if \quad\quad x_t \in X_S. \end{cases} \tag{5}$$

Note that (5) is used to describe the relationship between each model F_j and the corresponding training dataset X_j. We will introduce how to use these models for online monitoring in Section 3.3. The hierarchical scheme is depicted in Fig. 2 to clearly show how to obtain datasets X_j and how to conduct each iteration. For a given training data set X_1, a model F_1 is first constructed, and the monitoring index ϕ for all data is calculated accordingly. If there exist continuous n_c monitoring indices exceeding the threshold, i.e., $G(t) = 1$, then data with $I(t) = 1$ are picked out for the next iteration. When all $G(t) = 0$, the iteration is terminated.

2.2 Problem statements

According to (2)-(5), the calculation of monitoring indices is important, significantly impacting the modeling process and monitoring performance of our method. Certainly, model accuracy is also vital to compute a precise monitoring index. Hence, there are two key points in the hierarchical scheme: One is to determine the model F. Another is to define the monitoring index ϕ with corresponding threshold ϕ_{th}. Furthermore, there are some hyperparameters like n_c and those related to modeling. We describe exactly how to adjust them in Section 3.

3. HIERARCHICAL SCHEME FOR MULTIMODE PROCESS MONITORING

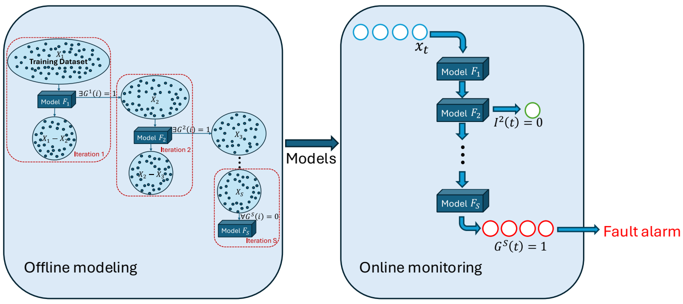


Fig. 3. Framework of the proposed hierarchical monitoring scheme.

We first plot Fig. 3 to show the proposed hierarchical scheme for multimode dynamic process monitoring. It mainly includes two parts: Offline modeling and online monitoring. For offline modeling, according to Fig. 2 and problem statements, model identification and monitoring index design are two crucial steps. In this paper, we use the well-known autoregressive (AR) model and introduce our monitoring method as follows.

3.1 Model identification

The AR model is a type of linear time series model that represents the current value of a variable as a linear combination of its past values. For a training dataset, an AR model of order p can be expressed as:

$$x_t = \theta_1 x_{t-1} + \theta_2 x_{t-2} + \cdots + \theta_p x_{t-p} + e_t, \tag{6}$$

where $\theta_1, \theta_2, \ldots, \theta_p$ are model parameters to be estimated, and e_t is a white noise error term with mean zero and constant variance.

Identifying an AR model involves two main steps: Selecting the appropriate model order p and estimating the model parameters $\theta_1, \theta_2, \ldots, \theta_p$. We use Akaike Information Criterion (AIC) for order selection in this paper and apply the ordinary least squares (OLS) for parameter estimation.

It should be noted that in our hierarchical scheme, the model is not limited to the AR model, we choose AR model owing to its simplicity and computational efficiency. Other models such as state-space models and exponential smoothing models can also be used depending on specific scenarios.

3.2 Monitoring index design

Based on (6), we use ϕ_t, a combined index defined by applying PCA on e_t, as the monitoring index:

$$\phi_t = T_t^2 + g^{-1} Q_t, \tag{7}$$

where T_t^2, Q_t are Hotelling's T^2 and squared prediction error (SPE) for e_t respectively. The g is a specific ratio related to the variance of residual components. The threshold calculation of ϕ_t has been described in Dong and Qin (2020). In this monitoring index calculation, only the number of principal components l should be determined, and this determination follows a certain criterion called cumulative percent variance (CPV). One can select the number of principal components such that the CPV reaches 95%.

Finally, for monitoring, the normal and abnormal data are determined by the following rules

$$\begin{cases} \phi_t \leq \phi_{th} \Rightarrow normal \\ \phi_t > \phi_{th} \Rightarrow abnormal \end{cases}, t = 1, \ldots, N. \tag{8}$$

3.3 Online monitoring

According to our hierarchical scheme, S AR models are utilized for online monitoring where each online data is input to each AR model sequentially and its prediction error e_t is used for monitoring index calculation. If there exists a monitoring index of AR model j that is not larger than the threshold, i.e., $I^j(t) = 0$, then the data point is normal. Otherwise, the data is abnormal. If n_c consecutive data are considered abnormal, i.e., $G(t) = 1$, then a fault is detected. The proposed hierarchical scheme for both offline training and online monitoring is summarized in Table 1.

Note that besides the model order p, the number of consecutive threshold exceedances n_c should also be determined by users. On the one hand, a large n_c is conservative, leading to a large delay in fault detection. On the other hand, a small n_c may result in a high false alarm rate (FAR). It requires all models to be sensitive to all variations, even for the noise. Balancing sensitivity (quickly detecting faults) and accuracy (avoiding false alarms) is important in this part. Besides the expertise, a feasible way to determine n_c is to set $n_c = 1$ first, and then count maximum

Table 1. Procedures of the proposed hierarchical scheme.

Offline training:	
Initialization:	Training dataset X_1, model number $S = 1$.
Step 1:	Construct an AR model F_S based on dataset X_S.
Step 2:	Calculate the monitoring index ϕ_t for each data $x_t \in X_S$ and corresponding threshold $\phi_{S,th}$.
Step 3:	According to (4), if there exists $G(t) = 1$, data with $I(t) = 1$ are selected to compose X_S after $S = S + 1$, and then go to step 1. If for all data $G(t) = 0$, the offline training ends.
Outputs:	AR models $F_1, \dots, F_S$ and thresholds $\phi_{1,th}, \dots, \phi_{S,th}$.
Online monitoring:	
Initialization:	AR models $F_1, \dots, F_S$, thresholds $\phi_{1,th}, \dots, \phi_{S,th}$, model counter $c = 1$ and threshold exceedance counter $C(t) = 0$.
Step 1:	Collect an online data x_t.
Step 2:	Input the data x_t to the AR model F_c, and compute the monitoring index ϕ_t.
Step 3:	If $I(t) = 0$, $t = t + 1$ and go to step 1 with $C(t) = 0$. If $I(t) = 1$ and $c < S$, then $c = c+1$ and go to step 2. Otherwise, $C(t) = C(t-1) + 1$ and go to step 4.
Step 4:	If $G(t) = 1$, the fault alarm is given. Otherwise, $t = t + 1$ and go to step 1.
Outputs:	Monitoring indices ϕ of each online data with corresponding thresholds ϕ_{th}.

conservative threshold exceedances after identifying each AR model. Finally, determine n_c based on the sensitive tolerance of users. We recommend practitioners try $n_c = 5$ according to our experiments.

4. APPLICATION TO TENNESSEE EASTMAN PROCESS

The proposed method is applied to TEP for verification. To demonstrate the superiority of the proposed hierarchical structure, we employ two multi-model monitoring approaches using mode partition: One is to use variational Bayesian Gaussian mixture model (VBGMM) for grouping data and canonical correlation analysis (CCA) for modeling and monitoring, which is called VBGMM-CCA proposed by Jiang and Yan (2019). Another is a variant of VBGMM-CCA which is VBGMM-AR. As the name implies, in VBGMM-AR, we apply AR model with PCA for modeling and monitoring.

4.1 Process description

The TEP is widely recognized as a benchmark for research in industrial process monitoring. This process involves producing two liquid products G and H, from four gaseous reactants: A, C, D and E, along with an inert component B, and an undesired byproduct F. Originally introduced by Downs and Vogel (1993), the TEP has since become a standard for studies on process monitoring. The process operates across six modes, each associated with specific target mass ratios and production rates. Ricker (1995) provided an overview of the optimal steady-state conditions for these six modes. In this study, we refer to Ricker's work and select four operating modes for our simulations, using the updated TEP model from Bathelt et al. (2015). Setpoints for each mode are detailed in Table 2.

We created three types of datasets: training data, normal test data, and test data with simulated faults. For training, we collected 10000, 13000, 5000 and 7000 samples for each mode respectively. The four modes alternate in order: mode 3 → mode 1 → mode 2 → mode 4 → mode 1 → mode 2. In total, we utilized 49 variables, including 8 manipulated and 41 process variables as outlined in Downs and Vogel (1993), for modeling. Note that there are 12 manipulated variables in TEP. Since four of them, which are recycle flow, reactor level, reactor temperature, and product separator level, do not vary across modes, they were excluded. For test data collection, each test data set includes 25000 datapoints where the mode transition follows mode 3 → mode 1 → mode 2 → mode 4. The fault test data is summarized in Table 3.

Table 2. Setpoints of four modes.

Setpoint label	Mode 1	Mode 2	Mode 3	Mode 4
Production	22.89	22.73	18.04	36.04
Stripper level	50	50	50	50
Separator level	50	50	50	50
Reactor level	65	65	65	65
Reactor pressure	2800	2800	2800	2800
Mol % G	53.8	11.66	90.09	53.35
yA	63.137	64.196	62.11	61.94
yAC	51	54.24	47.43	58.76
Reactor temperature	122.9	124.2	121.9	128.2
Recycle valve position	1	1	77.62	1
Steam valve position	1	1	1	1
Agitator setting	100	100	100	100

4.2 Method application

In our method, there are three hyperparameters to be determined. Firstly, as described in Section 3, we use AIC to set model order as $p = 5$. Then, PCA is applied to the prediction error e_t. The number of principal components is set to 20 with a CPV reaching 95%. The last hyperparameter is the tolerated number of consecutive exceedances, denoted as n_c. Initially, we set $n_c = 1$ and then observe the maximum consecutive threshold exceedances, which are 26 for AR model 1 and 5 for AR model 2. Since 5 is within a tolerable range, we determine $n_c = 5$ and obtain two AR models. Note that we construct dynamic models to extract dynamic patterns of the process rather than operating modes. Therefore, although there are four operating modes, two dynamic models are sufficient to represent dynamic patterns revealed by the four modes.

For VBGMM-AR, the model order is the same as that of our method. VBGMM can automatically determine how many groups the data should be divided into according to data distributions rather than operating modes. Thus, the process is partitioned into three parts and three AR models are established. Similarly, in VBGMM-CCA, three CCA models are constructed by using 8 inputs and 41 outputs.

To compare the performance of different approaches, FAR and delay time (DT) are computed as

Table 3. Information about test data with different faults.

Fault label	Descriptions	Start point of the fault	Mode	Type
1	Cooling water inlet temperature of reactor	13000	2	Random variation
2	Cooling water inlet temperature of separator	13000	2	Random variation
3	Variation coefficient of heat transfer in reactor	13000	2	Random variation
4	Variation coefficient of heat transfer in condenser	7000	4	Random variation
5	Unknown	7000	4	Unknown
6	Unknown	7000	4	Unknown
7	A and C feed flow (stream 4)	7000	4	Random variation

$$FAR = \frac{N(x_{normal}|I=1)}{N(x_{normal})}, \tag{9}$$

$$t_{delay} = t_d - t_f, \tag{10}$$

where t_d is detection time and t_f is fault occurrence time. For all methods, we record the fault detection time after 5 consecutive indices exceeding thresholds. Note that the numerator of (9) is the number of false alarms counted when indices exceed thresholds. It is different from the way we record fault detection.

4.3 Results and analysis

FAR and DT of all methods are presented in Tables 4 and 5 separately. We mark the DT with * in the case that before t_f, a fault detection has already occurred, but for comparison, we only provide DTs recorded after the true fault occurs. In Table 4, the normal test dataset is labeled as 0. As shown, our hierarchical method achieves the best performance, exhibiting shorter DTs and lower FARs across nearly all fault cases. In contrast, methods using mode partition, VBGMM-AR and VBGMM-CCA, have relatively high FARs. This indicates that the hierarchical structure introduced in this paper is superior in capturing multimode dynamics compared to mode partition approaches. Additionally, regarding detection speed, the hierarchical scheme detects faults more quickly than the VBGMM-based methods.

To further illustrate differences in fault detection accuracy, we take fault 1 as an example and present the monitoring results of all methods in Fig. 4, where the red line indicates the onset of the fault. As shown, VBGMM-CCA inaccurately signals faults even before the fault actually occurs. This premature fault detection is due to the inappropriate use of the static CCA model to monitor dynamic processes. Fig. 4(a) clearly demonstrates numerous indices exceeding thresholds despite no fault occurrence, highlighting the unreliability of VBGMM-CCA. VBGMM-AR performs better than VBGMM-CCA owing to the dynamic AR modeling, but still results in multiple threshold violations before fault occurrence. By contrast, our hierarchical method effectively avoids this problem, achieving timely fault detection without generating misleading alarms.

Beyond improved FAR and faster, more accurate detection, the proposed hierarchical approach also requires fewer models and hyperparameters. Consequently, our method not only provides superior performance but is also easier to implement, which significantly reduces user effort.

Table 4. FARs of all methods on test data.

Label	H-AR	VBGMM-AR	VBGMM-CCA
0	0.0028	0.1257	0.1442
1	0.0034	0.0639	0.1442
2	0.0034	0.1215	0.1551
3	0.0034	0.0638	0.1551
4	0.0046	0.1187	0.1091
5	0.0046	0.1204	0.1091
6	0.0046	0.1187	0.1091
7	0.0046	0.1194	0.1091

Table 5. DTs (samples) of all approaches on fault test data.

Fault label	H-AR	VBGMM-AR	VBGMM-CCA
1	29	3455	29*
2	5894	2002	123*
3	181	236	95*
4	89	8022	6*
5	137	847	6*
6	342	367	6*
7	85	336	6*

5. CONCLUSION

In this study, we proposed a hierarchical approach for multimode dynamic process monitoring, addressing and improving the limitations of conventional multiple-model schemes. By constructing dynamic models iteratively in a hierarchical structure without explicit mode partition, our approach offers a concise framework for capturing and monitoring multimode dynamics. This approach reduces the need for hyperparameter tuning, which facilitates its promotion in different processes. The simulation on TEP demonstrates that the proposed method is more concise while achieving satisfying monitoring performance. Our future work will focus on the nonlinearity extension of the hierarchical scheme.

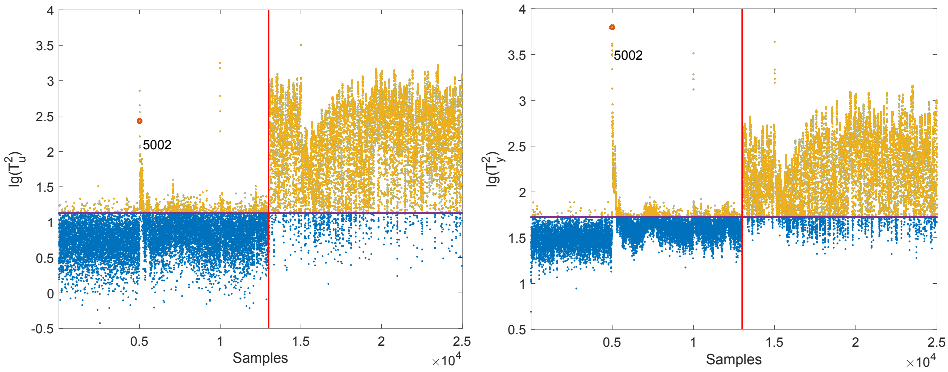
REFERENCES

Bathelt, A., Ricker, N.L., and Jelali, M. (2015). Revision of the tennessee eastman process model. *IFAC-PapersOnLine*, 48(8), 309–314. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

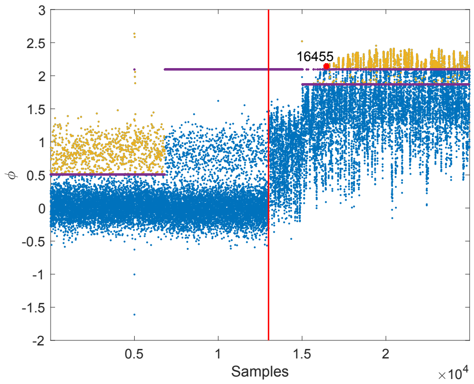
Chen, Z., Liu, C., Ding, S.X., Peng, T., Yang, C., Gui, W., and Shardt, Y.A.W. (2021). A just-in-time-learning-aided canonical correlation analysis method for multimode process monitoring and fault detection. *IEEE Transactions on Industrial Electronics*, 68(6), 5259–5270.

Dong, Y. and Qin, S.J. (2020). New dynamic predictive monitoring schemes based on dynamic latent variable models. *Industrial & Engineering Chemistry Research*, 59(6), 2353–2365.

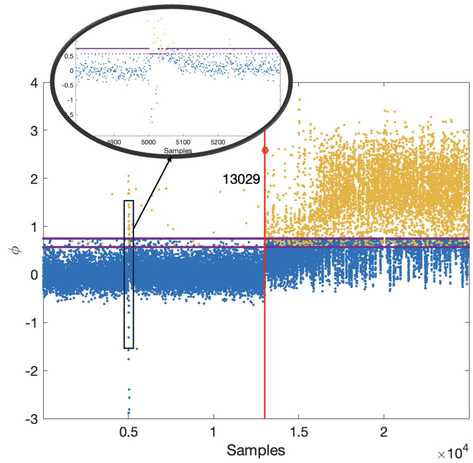
Downs, J. and Vogel, E. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3), 245–255. Industrial challenge problems in process control.



(a) VBGMM-CCA.



(b) VBGMM-AR.



(c) Proposed method.

Fig. 4. Monitoring results obtained by different approaches for the test data of fault 1.

Huang, K., Tao, Z., Liu, Y., Sun, B., Yang, C., Gui, W., and Hu, S. (2023). Adaptive multimode process monitoring based on mode-matching and similarity-preserving dictionary learning. *IEEE Transactions on Cybernetics*, 53(6), 3974–3987.

Jiang, Q. and Yan, X. (2019). Multimode process monitoring using variational bayesian inference and canonical correlation analysis. *IEEE Transactions on Automation Science and Engineering*, 16(4), 1814–1824.

Lyu, Y., Zhou, L., Cong, Y., Zheng, H., and Song, Z. (2024). Multirate mixture probability principal component analysis for process monitoring in multimode processes. *IEEE Transactions on Automation Science and Engineering*, 21(2), 2027–2038.

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

Ricker, N.L. (1995). Optimal steady-state operation of the tennessee eastman challenge process. *Computers & Chemical Engineering*, 19(9), 949–959.

Song, X.L., Zhang, N., He, Y.L., Xu, Y., and Zhu, Q.X. (2024). Cross-modality manifold adaptive network for industrial multimode processes and its applications. *IEEE Transactions on Automation Science and Engineering*, 1–10.

Wang, B., Li, Z., Dai, Z., Lawrence, N., and Yan, X. (2020). Data-driven mode identification and unsupervised fault detection for nonlinear multimode processes. *IEEE Transactions on Industrial Informatics*, 16(6), 3651–3661.

Wang, J., Song, C., Zhao, J., and Peng, X. (2023). A dynamic model-level method for multimode process monitoring using ω-metric. *IEEE Transactions on Control Systems Technology*, 31(6), 2790–2804.

Wu, H. and Zhao, J. (2020). Self-adaptive deep learning for multimode process monitoring. *Computers & Chemical Engineering*, 141, 107024.

Xu, P., Liu, J., Yu, F., Guo, Q., Tan, S., and Zhang, W. (2023). Toward multimode process monitoring: A scheme based on kernel entropy component analysis. *IEEE Transactions on Instrumentation and Measurement*, 72, 1–15.