Dynamical Component Extraction based Fault Detection for Industrial IoT With Application to Ironmaking Process

Ping Wu, Yicheng Yu, Xujie Zhang, Siwei Lou, Jinfeng Gao, Qian Zhang *Member, IEEE*, Chunjie Yang *Senior Member, IEEE*

***Abstract*—The Industrial Internet of Things (IIoT) has become a crucial infrastructure in the process industry, particularly in the era of Industry 4.0. Ensuring operational safety in industrial processes necessitates fault detection techniques, which play a pivotal role in IIoT systems. These systems continuously collect high-dimensional process data, which often exhibit dynamic behavior due to the inherent complexity of industrial operations. Consequently, the dynamic characteristics of such data pose significant challenges for fault detection. As a powerful dimensionality reduction technique, Dynamical Component Analysis (DyCA) decomposes multivariate measurements of a dynamical system into a deterministic component which can be described by a system of differential equations and independent noise components. DyCA incorporates the covariance matrices of both the signals, and their derivative, as well as their cross-correlation. By doing so, it identifies a low-dimensional subspace that minimizes the error in the underlying ordinary differential equations. The DyCA components are estimated to capture low-dimensional trajectories that characterize the process dynamics. This study proposes a novel data-driven fault detection method based on dynamical component analysis for dynamic processes. Leveraging these DyCA components that represent the low-dimensional trajectories to describe the process dynamics, Hotelling's T^2 and Square Prediction Error (SPE) statistics are utilized as monitoring metrics for fault detection. Case studies on the widely utilized Tennessee Eastman process benchmark and a real-world blast furnace ironmaking process are conducted to demonstrate the effectiveness and capability of the proposed DyCA based fault detection method, comparing its performance with other relevant methods.**

***Index Terms*—Industrial Internet of Things, Fault Detection, Dynamical Component Analysis, Dynamic Process, Blast Furnace Ironmaking Process.**

I. Introduction

TECHNOLOGICAL advancements in major industrial sectors such as power, minerals and manufacturing, are generating unprecedented volumes of data [1]. For example, the ironmaking process generates vast volumes of data from diverse sources, including programmable logic controllers (PLCs), industrial sensors, and local indicators. As highlighted, a single blast furnace, equipped with thousands of measurement points, can produce terabytes of data daily. However, these vast volumes of industrial data often remain largely untapped, transitioning from a potential asset to an operational burden for manufacturing plants. This massive, continuously streaming data source is a defining characteristic of the Industrial Internet of Things (IIoT), representing a paradigm shift beyond traditional, limited-scope time-series analysis [2], [3].

To manage the volume and velocity of this data, which typically exceeds the computational capacity of on-premise industrial IT infrastructure, the proposed platform leverages cloud computing. The data storage layer, architected on Hadoop, provides scalable and bidirectional storage for this big data environment. The model computing layer utilizes the Spark distributed processing engine on virtual machines, delivering the on-demand, high-performance computing necessary for training and executing sophisticated data-driven models. Within this cloud platform, data-driven applications for fault detection, diagnosis, and process optimization are developed, deployed, and managed as industrial microservices. The results and subsequent control commands are then communicated back to the factory floor, facilitating remote monitoring and control which is a core functionality of IIoT [4].

Fault detection techniques play a critical role in ensuring operational safety and maintaining production quality in the IIoT [5]–[7]. These techniques are generally categorized into model-based and data-driven approaches. Unlike model-based methods, data-driven approaches do not require an accurate mathematical model, making them especially advantageous for managing the complicated characteristics inherent in industrial processes [8], [9]. With the rapid advancements in sensor technology, data storage capabilities, and computational power, data-driven approaches have garnered significant attention from both industrial practitioners and the academic community [10], [11].

Among data-driven fault detection methods, multivariate statistical process monitoring (MSPM) techniques have been widely adopted due to their simplicity and reduced design effort compared to model-based approaches [12]. MSPM methods, which leverage both input and output information from the process, are currently prevalent in fault detection and diagnosis applications [13]. Notably, their ability to handle highly correlated variables highlights a significant advantage of these methods [14].

Principal component analysis (PCA) has been widely rec-

Ping Wu and Jinfeng Gao are with the School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou, China (e-mail: pingwu@zstu.edu.cn). Ping Wu and Jinfeng Gao are also with Zhejiang Key Laboratory of Digital Fashion and Data Governance, Zhejiang Sci-Tech University. Yicheng Yu is with the School of Mechanical Engineering, and the School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou, China.

Xujie Zhang, Siwei Lou and Chunjie Yang are with the College of Control Science and Engineering from Zhejiang University, Hangzhou, China.

Qian Zhang is Department of Electronics and Electrical Engineering Liverpool John Moores University, Liverpool, U.K.

Manuscript received April 19, 2021; revised August 16, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62573387, the Natural Science Foundation of Zhejiang province, China under Grant LY24F030004, the Fundamental Research Funds of Zhejiang Sci-Tech University(25222139-Y). (Corresponding author: Ping Wu.)

ognized as a powerful tool for fault detection in multivariate analysis. PCA is a well-established unsupervised technique for data dimensionality reduction, maximizing variance in process variables [15]. Wise et al. [16] provided the theoretical foundation for using principal component models to monitor multivariate processes, demonstrating that PCA facilitates monitoring when there are more measurements than process states. Zheng et al. [17] developed deep residual PCA by employing layer-by-layer information transformation upon the residual subspace of the deep model. Fan et al. [18] proposed a distributed monitoring method with integrated probability principal component analysis and minimal redundancy maximum relevance. Since most industrial processes are inherently dynamic, Ku et al. [19] extended static PCA to dynamic PCA (DPCA) by augmenting variables with time-lagged vectors to capture process dynamics. Huang and Yan [20] further integrated DPCA, dynamic independent component analysis, and Bayesian inference to improve dynamic process monitoring. In addition, de Andrade Melani et al. [21] proposed a hybrid framework combining moving window PCA and Bayesian networks for fault detection and diagnosis in dynamic systems. However, DPCA has several limitations. First, as the number of lags increases, both the dimensionality of the loading vectors and the number of parameters expand significantly. Second, the extracted latent variables focus solely on variance, without emphasizing dynamic content, making model interpretation challenging. To address these issues, Li et al. [22] developed an autoregressive PCA algorithm, incorporating a vector autoregressive (VAR) model to extract latent variables, providing a more explicit representation of dynamic relationships and a compact data structure model. Zhou et al. [23] extended autoregressive PCA to the autoregressive dynamic latent variable model, which captures both dynamic and static relationships. However, the VAR model used in autoregressive PCA is inconsistent with the maximum variance objective, as collinear static relationships can dominate a latent variable that prioritizes variance, failing to effectively capture dynamic relationships. To overcome these limitations, Dong et al. [24] introduced dynamic inner PCA (DiPCA), which explicitly extracts a set of dynamic latent variables that capture the most significant dynamic variations in the data. By selecting a given number of latent factors, DiPCA generates a subspace of principal time series that are most predictable based on their past data, offering geometric insights into the new dynamic model structure.

Different from PCA, slow feature analysis (SFA) extracts slowly varying latent variables from temporal data and, as a biologically inspired approach, was initially applied to the analysis of self-organization in complex-cell receptive fields using synthetic image sequences [25]. SFA has garnered increasing attention, with applications emerging across various fields such as nonlinear blind source separation, human action recognition, and remote sensing [26], [27]. Unlike PCA, the latent variables in SFA, referred to as slow features (SFs) are assumed to vary slowly over time and can be sequentially ordered based on their degree of slowness, which is statistically measurable. Mathematically, SFA allows for a separate description of the steady-state distribution and temporal variation distribution, offering improved interpretability over classical latent variable models in terms of temporal coherence. Shang et al. [28] developed an SFA-based fault detection strategy to distinguish real faults, characterized by dynamic anomalies, from normal deviations in operating conditions. Zhang and Zhao [29] introduced multiple SFA-based local models to describe the static variations captured by the steady-state distribution, enabling identification of steady deviations from normal operating conditions across different phases in batch process monitoring. Ma et al. [30] proposed a multistep dynamic SFA approach, which divides the monitoring process into multiple steps, separately monitoring the static component and residual space to optimize detection rates and extract more detailed information. Vishal Rishi and Tangirala [31] proposed probabilistic adaptive SFA, which accommodates measurement and process noise as well as nonlinear generative models for state estimation and classification.

Dimensionality reduction for multivariate time-series data is crucial, particularly in scenarios where the number of sensors exceeds the intrinsic dimensionality of the system. This situation frequently arises in industrial applications, such as chemical processing or blast furnace ironmaking, where numerous sensors are employed to monitor systems characterized by fundamentally low-dimensional dynamics. Recently, a novel approach known as Dynamical Component Analysis (DyCA) has been developed for dimensionality reduction. DyCA is particularly effective in scenarios where signals are composed of linearly independent deterministic sources combined with noise, and the underlying signal model is predominantly governed by linear rather than nonlinear differential equations. Under these conditions, DyCA can identify the relevant subspace of the signal, functioning as a blind source separation technique [32]. The method is based on the assumption that multivariate measurements of a dynamical system can be decomposed into a deterministic component, which can be described by a system of differential equations, and independent noise components. DyCA achieves this by decomposing a multivariate signal into time-dependent amplitudes and their corresponding multivariate modes, estimating the amplitudes in a manner that optimally satisfies a set of coupled ordinary differential equations (ODEs). Unlike PCA and SFA, which rely on stochastic modeling assumptions, DyCA is particularly well-suited for dimensionality reduction in multivariate time-series data with prominent deterministic dynamics. The primary objective of DyCA is to identify low-dimensional subspaces that effectively capture the system's dynamics while accounting for noise. To the best of the authors' knowledge, DyCA has not yet been applied to fault detection.

Motivated by the above discussions, a novel data-driven fault detection method is developed by introducing dynamical component analysis for IIoT. Using DyCA, process data are decomposed into two distinct parts: deterministic components (i.e., DyCA components) with reduced dimensionality, and stochastic components (i.e., noise components). Based on these decomposed components, Hotelling's T^2 and square prediction errors (SPE) are used as monitoring statistics for fault detection. The main contributions and novelty of this

paper are included as follows,

- To capture the process dynamics, dynamical component analysis is introduced to decompose the process data into multivariate modes and corresponding time-dependent amplitudes, which obey a set of coupled ordinary differential equations. The process dynamics is well described by extracting the low-dimensional trajectories.
- Based on the established deterministic components and noise components, two monitoring statistics including Hotelling's T^2 and square prediction error are established for fault detection. A novel data-driven fault detection is developed based on dynamical component analysis for dynamic processes.
- Case studies on the popular industrial Tennessee Eastman Process benchmark and a real-world blast furnace ironmaking process are conducted, demonstrating the superiority of the proposed DyCA based fault detection approach in comparison with related methods.

The rest of this paper is organized as follows. In Section II, the proposed DyCA based fault detection is demonstrated in details. Section III is devoted to the case studies. Conclusions are drawn in the final section.

II. Proposed DyCA based Fault Detection

A. DyCA

Given that the n-dimensional time series $\mathbf{x} \in \mathbb{R}^n$ with N samples are collected, the tth $\mathbf{x}$ can be decomposed into deterministic components and noise components,

$$\mathbf{x}(t) = \sum_{i=1}^{m} y_i(t)\omega_i + \sum_{i=1}^{p} e_i(t)\gamma_i \tag{1}$$

where $y_i(t)$ is the deterministic amplitude, ω_i are linearly independent. $e_i(t)$ is the noise amplitude, γ_i are linearly independent. And $m + p \leq n$.

Based upon the assumption that the $l(l \geq \frac{m}{2})$ deterministic amplitudes $y_i(t), i = 1, \ldots, l$ are governed by a set of linear ordinary differential equations,

$$\begin{cases} \dot{y}_1(t) = \sum_{i=1}^{m} a_{1,i} y_i(t) \\ \dot{y}_2(t) = \sum_{i=1}^{m} a_{2,i} y_i(t) \\ \quad\quad \vdots \\ \dot{y}_l(t) = \sum_{i=1}^{m} a_{l,i} y_i(t) \end{cases} \tag{2}$$

where $a_{1,1}, a_{1,2}, \ldots, a_{l,m}$ are the coefficients of linear ODEs. Moreover, the coefficients can be written in matrix form $\mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2]$,

$$\mathbf{A}_1 = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ a_{2,1} & \cdots & a_{2,m} \\ \vdots & \ddots & \vdots \\ a_{l,1} & \cdots & a_{l,m} \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} a_{1,l+1} & \cdots & a_{1,m} \\ a_{2,l+1} & \cdots & a_{2,m} \\ \vdots & \ddots & \vdots \\ a_{l,l+1} & \cdots & a_{l,m} \end{bmatrix}$$

And the remained deterministic amplitudes $y_i(t), i = l+1, \ldots, m$ are governed by a set of nonlinear ordinary differential equations,

$$\begin{cases} \dot{y}_{l+1}(t) = f_{l+1}(y_1(t), y_2(t), \ldots, y_m(t)) \\ \dot{y}_{l+2}(t) = f_{l+2}(y_1(t), y_2(t), \ldots, y_m(t)) \\ \quad\quad \vdots \\ \dot{y}_m(t) = f_m(y_1(t), y_2(t), \ldots, y_m(t)) \end{cases} \tag{3}$$

where $f_{l+1}, f_{l+2}, \ldots, f_m$ are unknown, non-linear, smooth functions of nonlinear ODEs.

Eq.(1) can be rewritten in matrix form as follows,

$$\mathbf{X} = \mathbf{\Omega}\mathbf{Y} + \mathbf{\Gamma}\mathbf{E} \tag{4}$$

where $\mathbf{X} = \begin{bmatrix} \mathbf{x}(1) & \mathbf{x}(2) & \ldots & \mathbf{x}(N) \end{bmatrix} \in \mathbb{R}^{n \times N}$, $\mathbf{\Omega} = \begin{bmatrix} \omega_1 & \omega_2 & \ldots & \omega_m \end{bmatrix} \in \mathbb{R}^{n \times m}$. And $\mathbf{Y}$ is denoted,

$$\mathbf{Y} = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(N) \\ y_2(1) & y_2(2) & \cdots & y_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ y_m(1) & y_m(2) & \cdots & y_m(N) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix} \in \mathbb{R}^{m \times N}$$

The noise components and its coefficient matrices $\mathbf{\Gamma}$ and $\mathbf{E}$ are formed similarly.

By the facts that the deterministic amplitudes $y_i(t)$ obey by a set of ODE, then the goal of DyCA is to extract the deterministic part $\mathbf{\Omega}\mathbf{Y}$ and stochastic part $\mathbf{\Gamma}\mathbf{E}$ from the signal $\mathbf{X}$ to capture the underlying dynamics of the ODE system. Denote the generalized left inverse of $\mathbf{\Omega}$ as $\mathbf{\Omega}^{\ddagger}$ where $\mathbf{\Omega}^{\ddagger}\mathbf{\Omega} = \mathbf{I}_m$. Since $\mathbf{\Omega}$ is composed of linearly independent vectors, then the rows $\mathbf{\Omega}^{\ddagger}$ consists of a set of linearly independent vectors $\{\alpha_1^\top, \alpha_2^\top, \cdots, \alpha_m^\top\}$.

The amplitude $y_i(t)$ and its derivative $\dot{y}_i(t)$ are then calculated as,

$$\begin{cases} y_i(t) = \mathbf{x}(t)^\top \alpha_i \\ \dot{y}_i(t) = \dot{\mathbf{x}}(t)^\top \alpha_i \end{cases} \tag{5}$$

To seek $\mathbf{\Omega}^{\ddagger}$, following the Eq.(2), it can obtain,

$$\dot{\mathbf{x}}(t)^\top \alpha_i = \sum_{k=1}^{m} a_{i,k} \mathbf{x}(t)^\top \alpha_k = \mathbf{x}(t)^\top \beta_i \tag{6}$$

where $\beta_i = \sum_{k=1}^{m} a_{i,k}\alpha_k$ for $i = 1, \cdots, l$.

To estimate the projecting vectors α_i and β_i, then least squares approach is employed for data with additional noise. The least squares problem is defined,

$$\begin{aligned} &\min_{\{\alpha_i, \beta_i, i=1,\cdots,l\}} \sum_{i=1}^{l} \frac{\|\dot{\mathbf{x}}(t)^\top \alpha_i - \mathbf{x}(t)^\top \beta_i\|^2}{\|\dot{\mathbf{x}}(t)^\top \alpha_i\|^2} \\ &\text{s.t. } \alpha_i^\top \alpha_j = 0, i \neq j. \end{aligned} \tag{7}$$

In [32], the optimization problem Eq.(8) is solved by employing generalized eigenvalue decomposition (GED). The correlation matrices are first defined,

$$\begin{cases} \Sigma_0 = \frac{1}{N}\mathbf{X}\mathbf{X}^\top \\ \Sigma_1 = \frac{1}{N}\dot{\mathbf{X}}\mathbf{X}^\top \\ \Sigma_2 = \frac{1}{N}\dot{\mathbf{X}}\dot{\mathbf{X}}^\top \end{cases} \tag{8}$$

Then, a generalized eigenvalue problem is established for estimating the projecting vectors α_i,

$$\Sigma_1 \Sigma_0^{-1} \Sigma_1^\top \alpha_i = \lambda \Sigma_2 \alpha_i \tag{9}$$

Based on the estimates α_i, the amplitudes $y_i(t)$ can be obtained,

$$y_i(t) = \mathbf{x}(t)^\top \alpha_i \tag{10}$$

Furthermore, β_i is derived as,

$$\beta_i = \Sigma_0^{-1} \Sigma_1^\top \alpha_i \tag{11}$$

By sorting the eigenvalues λ_i and defining a threshold value σ, then the number l is selected as the number of λ_i that are larger than σ.

With the assumption that the matrix $\mathbf{A}_2$ of the ODE coefficient matrix has full rank $m - l$ and the definition of $\beta_i = \sum_{k=1}^{m} a_{i,k} \alpha_k$ for $i = 1, \cdots, l$, it can obtain,

$$m := \dim(span(\alpha_1, \cdots, \alpha_l, \beta_1, \cdots, \beta_l)) \tag{12}$$

Then, the $\mathbf{\Omega}^\ddagger$ is constructed by choosing a minimal subset of vectors β_i (i.e., $\{\beta_{k_1}, \cdots, \beta_{k_{m-l}}\}$) which are linearly independent to all α_i,

$$\mathbf{\Omega}^\ddagger = \left[\alpha_1, \cdots, \alpha_l, \beta_{k_1}, \cdots, \beta_{k_{m-l}}\right] \tag{13}$$

Consequently, the estimates of $\mathbf{y}_i$ are given,

$$\mathbf{Y} = \mathbf{\Omega}^\ddagger \mathbf{X} \tag{14}$$

To estimate the DyCA components $\mathbf{\Omega}$, the least squares problem is solved as,

$$\min_{\mathbf{\Omega}} \|\mathbf{X} - \mathbf{\Omega}\mathbf{Y}\|_F^2 \tag{15}$$

By defining $\mathbf{\Sigma}_\mathbf{Y} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^\top$, the $\mathbf{\Omega}$ is estimated,

$$\mathbf{\Omega} = \frac{1}{N}\mathbf{X}\mathbf{Y}^\top \mathbf{\Sigma}_\mathbf{Y}^{-1} \tag{16}$$

The algorithmic procedure of DyCA modelling is described in Algorithm 1.

Remark 1. The key hyperparameter in DyCA is the threshold σ, which determines the subspace dimension l by specifying the number of eigenvalues λ_i retained (where $\lambda_i > \sigma$). Acknowledging that a definitive theoretical basis for selecting σ is not established [32], a trial and error validation procedure is employed to determine its optimal value.

B. Relationship between DyCA, PCA and SFA

For PCA, the principal components are extracted by maximizing the variance of the process data. Thus, it should solve the following generalized eigenvector problem,

$$\Sigma_0 \alpha_{pca} = \lambda \alpha_{pca} \tag{17}$$

As shown in Eq.(17), PCA mainly involves the covariance matrix Σ_0, where the variance information of $\mathbf{x}$ is taken into consideration.

For SFA, the slow features are calculated through two-step generalized eigenvalue decomposition. The first decomposition is for whitening,

$$\Sigma_0 \alpha_{whitening} = \lambda_{whitening} \alpha_{whitening} \tag{18}$$

Then, the whitened vector $\mathbf{z}$ is obtained by

$$\mathbf{z} = \mathbf{Q}\mathbf{x} \tag{19}$$

where $\mathbf{Q}$ is the whitening transformation matrix calculating from Eq. (18). Furthermore, generalized eigenvalue decomposition is performed on the covariance matrix of the derivate of $\mathbf{z}$ to obtain the transformation matrix for slow feature extraction,

$$\Sigma_{\dot{\mathbf{z}}} \alpha_{slow} = \lambda_{slow} \alpha_{slow} \tag{20}$$

where $\Sigma_{\dot{\mathbf{z}}}$ is the covariance matrix of $\dot{\mathbf{z}}$.

Table I presents a comparison of PCA, SFA and DyCA, highlighting their respective methodologies and capabilities. PCA conducts generalized eigenvalue decomposition on the covariance matrix of $\mathbf{x}$, focusing solely on static features and neglecting the dynamic characteristics inherent in process data. SFA, on the other hand, performs a two-step GED: first on $\mathbf{x}$, and then on the derivative of $\mathbf{x}$, enabling it to effectively capture and evaluate process dynamics. In contrast to PCA and SFA, DyCA incorporates the covariance matrices of both $\mathbf{x}$ and its derivative, while also considering the correlation between $\mathbf{x}$ and its derivative. By minimizing the errors in the ordinary differential equations governing the dynamic system, DyCA provides an efficient framework for capturing dynamic behaviors in process data.

C. DyCA based fault detection

Given that N samples are collected under normal operation condition (NOC), the low-dimensional DyCA model can be established following the preceding subsection. Similar to PCA model, the DyCA model can be represented as,

$$\mathbf{x}(t) = \overbrace{(\mathbf{\Omega}^\ddagger)^\top \mathbf{y}(t)}^{deterministic\ part} + \underbrace{\epsilon(t)}_{stochastic\ part} \tag{21}$$

For the sample $\mathbf{x}(t)$, the low-dimensional projection data can be obtained using the projection matrix $\mathbf{\Omega}^\ddagger$,

$$\mathbf{y}(t) = \mathbf{\Omega}^\ddagger \mathbf{x}(t) \tag{22}$$

Based on the projection data $\mathbf{y}$, Hotelling's T^2 is established for fault detection,

$$T_\mathbf{y}^2 = \mathbf{y}(t)^\top \Sigma_\mathbf{y} \mathbf{y}(t) \tag{23}$$

Furthermore, square prediction error is built as follows,

$$SPE = \epsilon(t)^\top \epsilon(t) = \mathbf{x}(t)^\top (\mathbf{I} - (\mathbf{\Omega}^\ddagger)^\top \mathbf{\Omega}^\ddagger) \mathbf{x}(t) \tag{24}$$

To determine the upper control limit (UCL) for fault detection, the kernel density estimation (KDE) method [33] is utilized. KDE is a robust non-parametric technique for estimating the probability density function (PDF). This method is widely applied to address the violation of the Gaussian assumption in fault detection scenarios. The PDF of the DyCA T^2 is represented as follows,

$$P(T^2) = \frac{1}{N\mu} \sum\nolimits_{j=1}^{N} k(\frac{T^2 - T^2(j)}{\mu}) \tag{25}$$

Algorithm 1 Procedure of the DyCA modelling

Input :Data $\mathbf{X}$, threshold value σ.
Output: Transformation matrix $\mathbf{\Omega}^{\ddagger}$ and DyCA components $\mathbf{\Omega}$.
Compute: Correlation matrices of $\mathbf{x}$ and $\dot{\mathbf{x}}$ and Cross-correlation matrix between $\mathbf{x}$ and its derivate $\dot{\mathbf{x}}$ as in Eq.(8).
Solve: Generalized eigenvalue problem in Eq.(9) to derive projecting vectors α_i.
Estimate: Transformation matrix $\mathbf{\Omega}^{\ddagger}$ as in Eq.(13).
Estimate: Amplitudes $y_i(t)$ and DyCA components $\mathbf{\Omega}$ as in Eqs.(14) and (16).

TABLE I
COMPARISON OF PCA, SFA AND DYCA.

Model	GED problem	Dynamics	Correlation
PCA	$\Sigma_0 = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$	✗	✗
SFA	$\Sigma_0 = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$ & $\Sigma_{\dot{\mathbf{z}}} = \frac{1}{N}\dot{\mathbf{Z}}\dot{\mathbf{Z}}^\top$	✓	✗
DyCA	$\Sigma_1\Sigma_0^{-1}\Sigma_1^\top\Sigma_2^{-1}$	✓	✓

where μ is the kernel bandwidth, $T^2(j), j = 1, 2, \ldots, N$ are the samples of T^2. $k(\cdot)$ is the kernel function. A Gaussian kernel, which is commonly employed, is selected as the kernel function:

$$k(g) = \frac{1}{\sqrt{2\pi}} exp(-\frac{g^2}{2}) \tag{26}$$

Based on the estimated PDF, the UCL of the DyCA T^2 monitoring statistic, denoted as J_{T^2}, can be determined at a given significance level α as follows:

$$\begin{aligned} P(T^2 < J_{T^2}) &= \int_{-\infty}^{J_{T^2}} \frac{1}{N\mu}\sum_{j=1}^{N} k(\frac{T^2 - T^2(j)}{\mu})\mathrm{d}T^2 \\ &= \alpha \end{aligned} \tag{27}$$

Similarly, the UCL of the DyCA SPE monitoring statistic, denoted as J_{SPE}, is calculated using the same methodology. Fault detection is performed by comparing the monitoring statistics against their respective UCLs. A fault is detected when either $T^2 > J_{T^2}$ or $SPE > J_{SPE}$.

Data-driven fault detection typically involves two phases: an offline modeling phase and an online monitoring phase. The proposed DyCA based fault detection framework is illustrated in Fig. 1. The details of the proposed DyCA based fault detection method are described as follows:

- Offline training phase
  - Step 1. Normalize the collected training dataset under normal operation conditions.
  - Step 2. Build the DyCA model with Eqs.(8)-(15).
  - Step 3. Estimate the projection data and errors with Eqs.(21) and (22).
  - Step 4. Establish the monitoring statistics and thresholds with Eqs.(23-27).
- Online monitoring phase
  - Step 1. Normalize the collected new sample.
  - Step 2. Extract the projection data and errors with Eqs.(21) and (22).
  - Step 3. Estimate the monitoring statistics T^2 and SPE with Eqs.(23) and (24).
  - Step 4. Compare the online monitoring statistic T^2 and SPE to the corresponding UCLs J_{T^2} and J_{SPE}, and trigger alarms if $T^2 > J_{T^2}$ or $SPE > J_{SPE}$.

As illustrated in Fig. 1, during the offline phase, the DyCA model is trained using historical process data. In the online phase, real-time fault detection is achieved through four computationally efficient steps: (1) normalization of each newly collected sample; (2) projection of the sample into the low-dimensional subspace and computation of residuals using the trained DyCA model; (3) calculation of the monitoring statistics (T^2) and (SPE); and (4) comparison of these statistics with their corresponding control limits (J_{T^2}) and (J_{SPE}) to determine whether an alarm should be triggered. Since these steps only involve simple normalization, linear projection, and statistical calculation, the computational cost is low, ensuring real-time applicability.

In the offline phase, the DyCA model is trained by computing correlation matrices and solving a generalized eigenvalue problem of size $n \times n$, where n denotes the number of process variables. The overall computational complexity of this stage is approximately $\mathcal{O}(n^2N + n^3)$, with N being the number of training samples. Since this phase is performed only once, it does not affect real-time performance. In the online monitoring phase, each newly collected sample is normalized, projected into the low-dimensional subspace using the trained model, and the monitoring statistics (T^2 and SPE) are calculated. These operations require only basic matrix-vector multiplications, with a computational cost of $\mathcal{O}(nl)$, where l is the number of retained components. As a result, the online monitoring is extremely fast, enabling real-time fault detection and accurate traceability to the production process.

III. CASE STUDIES

In this section, the widely used Tennessee Eastman Process (TEP) industrial benchmark and a real blast furnace ironmaking process (BFIP) are utilized to further validate the effectiveness of the proposed DyCA-based fault detection method. For comparative analysis, representative approaches including PCA, DPCA [19], DiPCA [24], and SFA [28] are employed. All experiments are implemented in Python 3.8 and executed on a computing platform equipped with Windows 11, an NVIDIA RTX 3070 Ti GPU, a 12th Gen Intel® Core i9-12900H CPU, and 32 GB of RAM.

To quantitatively assess the performance of the proposed method against these benchmark methods, two performance

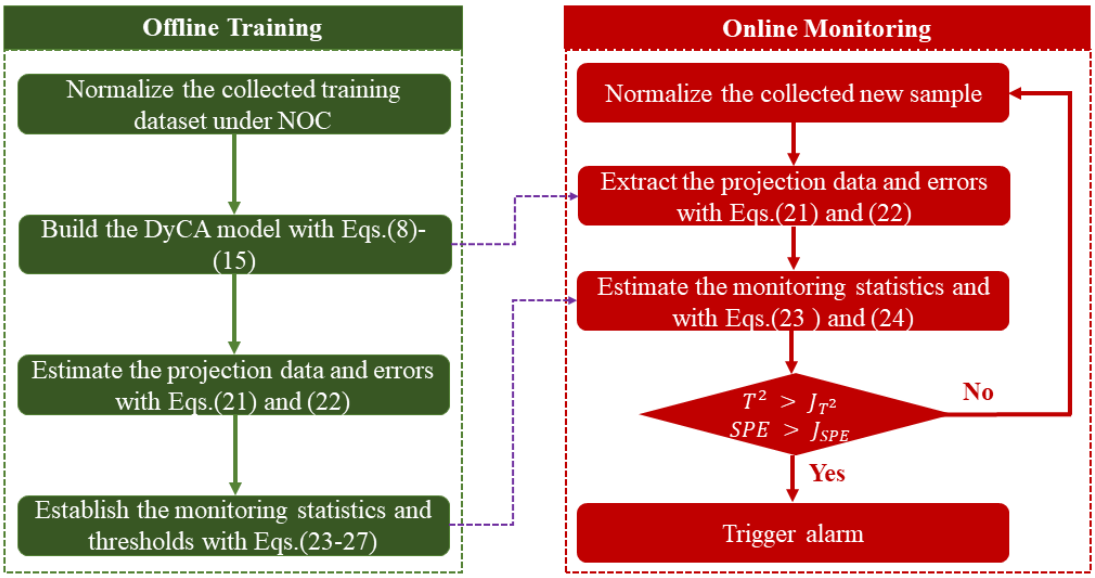

Fig. 1. Schematic of the proposed DyCA based fault detection method.

metrics are adopted: the fault detection rate (FDR) and the false alarm rate (FAR). These metrics are defined as follows:

$$FDR = \frac{N_{f,c}}{N_f} * 100\% \tag{28}$$

$$FAR = \frac{N_{n,ic}}{N_n} * 100\% \tag{29}$$

Here, N_f represents the total number of faulty samples, $N_{f,c}$ denotes the number of correctly detected faulty samples, N_n is the total number of normal samples, and $N_{n,ic}$ refers to the number of normal samples incorrectly classified as faulty.

A. Tennessee Eastman Process

The Tennessee Eastman Process is an open and challenging chemical simulation platform developed by Eastman Chemical Company, which is based on real chemical reaction processes [34]. A total of 21 predefined faults are included in the TEP dataset [34]. In this study, 22 measurement variables (XMEAS (1)–XMEAS (22)) and 11 manipulated variables (XMV (1)–XMV (11)) are selected for fault detection. The simulation data used in this study can be downloaded from the website http://web.mit.edu/braatzgroup/links.html.

For this case, 960 samples collected under normal operating conditions are used as the training dataset. The test dataset also consists of 960 samples for each fault scenario, with faults introduced at the 161st sample. For PCA, 20 latent variables are retained. For DPCA, a time lag of 1 is applied, retaining 20 latent variables. In the case of DiPCA, a time lag of 1 is used with 24 retained latent variables. For SFA, 20 slow feature vectors are retained, while in DyCA, 16 latent variables are selected. A confidence level of 0.99 is applied consistently across all comparison methods.

Table II summarizes the FDRs for 21 faults using different methods. As shown in Table II, DPCA T^2, DiPCA ϕ_v^2, SFA T^2, and DyCA T^2 outperform PCA T^2, as these methods incorporate process dynamics, which are not accounted for in PCA. Both DPCA and DiPCA utilize time-lagged models to address process dynamics, yet their performance remains inferior to that of SFA and DyCA. When comparing SFA with DyCA, the DyCA T^2 statistic achieves the highest average FDR, reaching 80.05%. For static residuals, it is observed that the SFA T_e^2 statistic demonstrates a relatively high average FDR of 70.87%. Table III presents the average false alarm rates for the 21 faults using different methods. From Table III, it is evident that both DiPCA and SFA exhibit relatively high average FARs. In contrast, the average FAR of DyCA T^2 is only 1.55%, which is deemed acceptable. In summary, the proposed DyCA method demonstrates superior fault detection performance compared to other methods, achieving both a high FDR and an acceptable FAR.

B. Blast Furnace Ironmaking Process

To further validate the effectiveness of the DyCA approach, practical experiments were conducted on a real-world BFIP. The blast furnace functions as the core unit of the BFIP, supported by five auxiliary subsystems: iron ore and coke feeding, pulverized coal injection, waste gas treatment, heated air supply, and tapping, as depicted in Fig. 2. In the BFIP, sintered ore, iron ore, and coke are fed into the blast furnace from the top, while pulverized coal and heated air are injected at the bottom through the tuyere apparatus. Within the furnace, high-temperature and high-pressure conditions facilitate complex chemical and physical reactions, resulting in the production of molten iron. Concurrently, waste gases are discharged from the top of the furnace and subsequently recycled for further utilization.

The IIoT platform deployed for the blast furnace ironmaking process at Liuzhou Iron & Steel Co., Ltd. is designed with a four-layer IoT system architecture, as illustrated in Fig. 3. At the Edge Layer, devices such as PLCs, DCS, MES,

TABLE II
FDRS FOR 21 FAULTS WITH DIFFERENT METHODS(%): TEP.

Fault	PCA		DPCA		DiPCA		SFA				DyCA	
	T^2	SPE	T^2	SPE	ϕ_v^2	ϕ_s^2	T^2	T_e^2	S^2	S_e^2	T^2	SPE
1	99.63	100.00	99.62	26.28	100.00	99.63	99.75	99.00	31.00	0.50	100.00	99.25
2	98.38	93.00	98.50	23.40	98.00	97.75	98.50	92.25	7.13	0.63	98.25	98.25
3	2.50	3.50	1.25	2.13	3.88	7.00	3.88	1.50	4.63	0.25	4.38	1.00
4	59.63	99.88	100.00	1.75	100.00	99.88	99.50	99.38	4.38	0.38	100.00	60.75
5	25.13	39.88	27.28	7.38	93.25	92.38	100.00	100.00	23.13	0.25	100.00	28.25
6	99.13	100.00	100.00	95.49	99.88	99.88	100.00	100.00	92.50	11.13	100.00	100.00
7	100.00	40.50	100.00	13.89	100.00	99.88	100.00	99.13	33.25	0.00	99.88	100.00
8	97.63	81.75	97.50	29.04	97.00	95.50	98.00	50.00	86.25	0.63	98.13	97.63
9	2.75	2.63	1.63	0.88	3.75	7.75	4.25	1.38	5.13	0.25	3.50	1.13
10	34.25	53.00	55.94	7.88	80.38	42.13	91.00	57.00	35.63	0.38	90.75	39.88
11	56.13	69.25	62.95	1.25	86.88	86.38	56.63	73.63	12.75	7.75	79.13	55.25
12	98.88	87.75	98.75	33.42	96.00	93.63	99.88	87.50	98.75	3.50	99.88	98.50
13	94.63	95.75	95.24	24.28	95.00	92.38	95.25	90.00	85.75	1.00	95.38	94.25
14	100.00	92.75	91.61	0.63	100.00	99.88	73.88	100.00	94.25	99.88	100.00	99.88
15	2.88	5.00	2.00	1.63	2.50	8.13	8.50	0.50	8.75	0.63	9.63	3.38
16	19.00	52.75	83.10	5.76	58.38	32.00	92.38	70.75	36.13	1.63	88.00	22.00
17	84.00	97.75	56.95	3.38	98.00	97.25	91.38	95.13	26.75	28.63	97.00	86.13
18	89.75	90.63	89.24	76.97	90.88	90.38	90.25	89.63	85.63	9.13	90.63	89.25
19	16.75	40.00	31.91	1.50	62.13	47.75	81.00	87.13	63.50	69.00	79.88	2.38
20	45.75	57.00	77.35	1.25	59.63	51.50	90.50	73.88	28.25	9.75	86.88	42.75
21	46.75	53.25	55.57	0.00	39.88	36.63	58.63	20.50	1.63	0.75	59.75	38.50
Ave	60.64	64.57	67.92	17.06	74.54	70.36	77.77	70.87	41.20	11.71	**80.05**	59.92

TABLE III
AVERAGE FARS FOR 21 FAULTS WITH DIFFERENT METHODS(%): TEP.

Fault	PCA		DPCA		DiPCA		SFA				DyCA	
	T^2	SPE	T^2	SPE	ϕ_v^2	ϕ_s^2	T^2	T_e^2	S^2	S_e^2	T^2	SPE
FAR	1.13	1.76	0.78	0.39	1.95	5.78	1.79	1.01	3.21	**0.18**	1.55	0.45

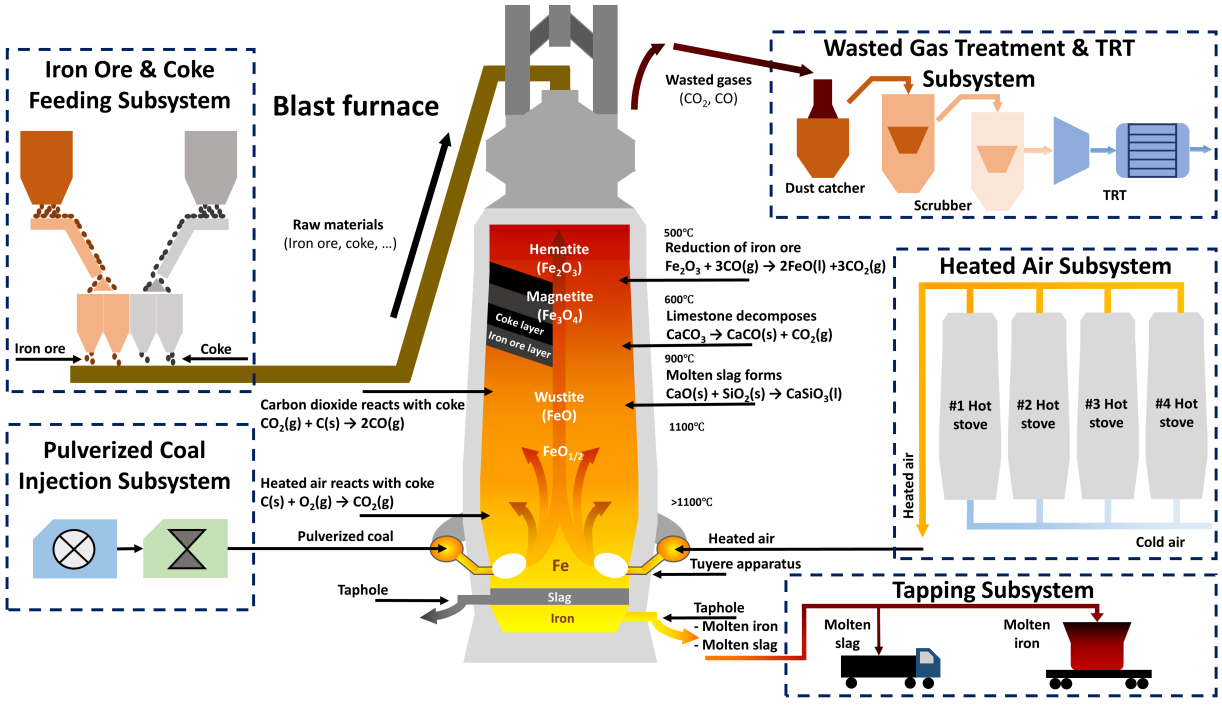

Fig. 2. The diagram of BFIP.

and LIMS continuously collect real-time operational data from the blast furnace, providing the raw data foundation for subsequent analytics. The Industrial IaaS Layer offers fundamental computing infrastructure, including servers, storage,

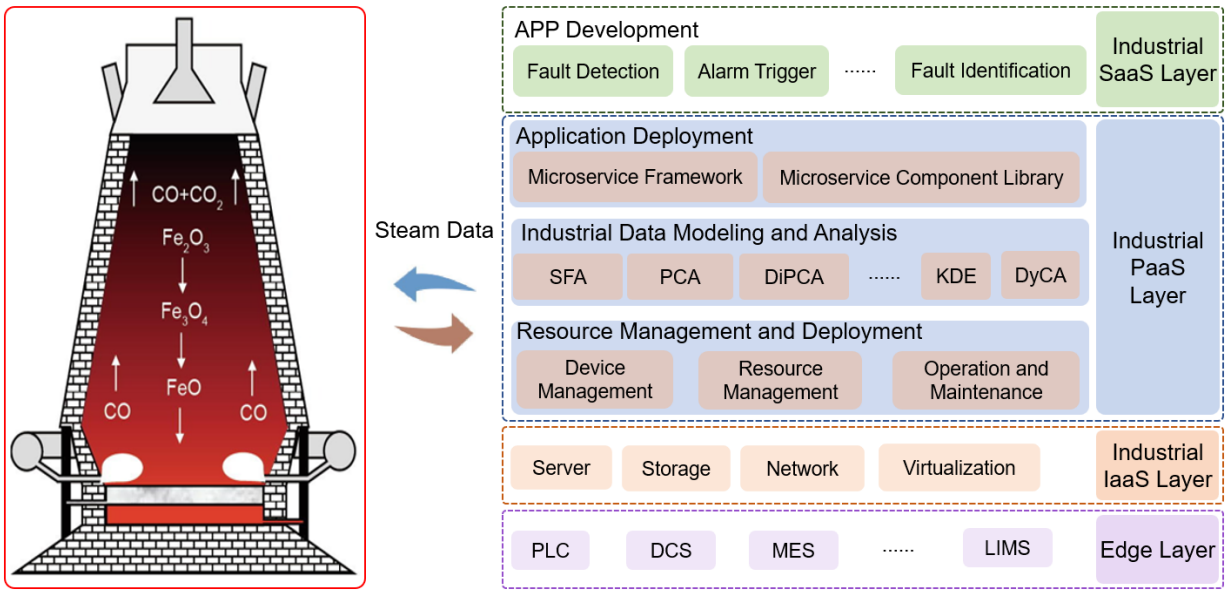

Fig. 3. Schematic architecture of the IIoT platform of the No 2. blast furnace at Liuzhou Iron & Steel Co. Ltd.

networking, and virtualization where Hadoop serves as the distributed computing and storage framework. The Hadoop Distributed File System (HDFS) ensures reliable and scalable data storage, while MapReduce enables parallel preprocessing of large-scale industrial data. Within the Industrial PaaS Layer, three functional modules operate synergistically: (1) Resource Management and Deployment manages devices, resources, and system maintenance to ensure stable operation; (2) Industrial Data Modeling and Analysis integrates algorithms such as SFA, PCA, DiPCA, KDE, and DyCA, with Spark's in-memory computing and rich ecosystem (including MLlib and GraphX) accelerating iterative modeling and analytical workflows; and (3) Application Deployment utilizes a microservice framework and component library for flexible and scalable deployment of industrial applications. Finally, the Industrial SaaS Layer provides specialized services, such as fault detection, alarm triggering, and fault identification that leverage the computational and analytical capabilities of the lower layers to support real-time monitoring and optimization of blast furnace operations.

This study considers two representative fault scenarios: hanging (denoted as Fault 1) and hot blast stove malfunction (denoted as Fault 2). In the hanging fault scenario, irregular movements of the burden are observed, where the descent of the burden may intermittently stop or suddenly drop by several meters. Such irregular behavior can locally affect the permeability of the burden and induce asymmetries within the blast furnace [35]. In the case of the hot blast stove malfunction, the hot blast valve becomes stuck, leading to an inadequate supply of hot air. To facilitate fault detection, 13 operational variables and 21 process variables were selected as input variables based on expert knowledge. The specific details of these operational and process variables are provided in Table IV.

In April 2021, a dataset of 1090 samples from normal operations was collected in collaboration with field engineers for model development. This dataset was partitioned into a training set (940 samples) and a separate validation set (150 samples). The validation set was used exclusively for hyperparameter tuning during the DyCA modeling process. For fault analysis, the dataset comprises real-world process anomalies, also identified with field engineers. For each fault scenario, a continuous time-series was constructed by concatenating 150 normal samples (immediately preceding the fault) with 150 subsequent fault samples. This creates a definitive transition at the 151st sample, enabling an unambiguous evaluation of detection algorithms at the onset of a real-world fault. The BFIP data that support the findings of this study are not openly available due to confidentiality agreements and are available from the corresponding author upon reasonable request.

Based on empirical observations and validation results, the parameters for the compared methods are configured as follows. For PCA, 20 latent variables are retained. For DPCA, a time lag of 2 is applied, retaining 8 latent variables. For DiPCA, a time lag of 2 is used, with 30 latent variables retained. For SFA, the number of slow features is set to 8,

TABLE IV
MEASURED OPERATIONAL AND PROCESS VARIABLES OF BFIP.

	No.	Description	No.	Description
Operational variables	OV1	Oxygen-enriched rate, %	OV8	Top pressure d, kPa
	OV2	Standard wind speed, m/s	OV9	Heat wind pressure a, kPa
	OV3	Oxygen-enriched volume, $10km^3/h$	OV10	Heat wind pressure b, kPa
	OV4	Cold wind volume, $10km^3/h$	OV11	Outlet wind speed, m/s
	OV5	Top pressure a, kPa	OV12	Heat wind temperature, ^oC
	OV6	Top pressure b, kPa	OV13	Coal injection setting, t/h
	OV7	Top pressure c, kPa		
Process variables	PV1	Gas permeability, $m^3/min \cdot kPa$	PV12	Total differential pressure, kPa
	PV2	CO content, %	PV13	Cold wind temperature, ^oC
	PV3	H_2 content, %	PV14	Top temperature a, ^oC
	PV4	CO_2 content, %	PV15	Top temperature b, ^oC
	PV5	Blast kinetic energy, kJ/s	PV16	Top temperature c, ^oC
	PV6	Belly gas volume, m^3/min	PV17	Top temperature d, ^oC
	PV7	Belly gas index, $m^3/(min \cdot m^2)$	PV18	Drop tube temperature, ^oC
	PV8	Adiabatic flame temperature, ^oC	PV19	Coal injection per hour, t
	PV9	Oxygen-enriched pressure, kPa	PV20	drag coefficient, /
	PV10	Cold wind pressure a, kPa	PV21	Coal injection in the previous hour, t
	PV11	Cold wind pressure b, kPa		

TABLE V
FDRS AND FARS FOR FAULT 1 AND FAULT 2: BFIP (%)

	Fault No.	PCA		DPCA		DiPCA		SFA				DyCA	
		T^2	SPE	T^2	SPE	ϕ_v^2	ϕ_s^2	T^2	T_e^2	S^2	S_e^2	T^2	SPE
FDR	1	70.00	59.30	74.67	30.00	77.33	56.67	86.67	87.33	14.00	18.67	**90.00**	28.00
	2	11.33	18.00	7.33	2.00	48.76	16.00	66.33	50.67	1.33	1.33	**72.70**	0.00
	Average	40.67	38.65	41.00	16.00	63.05	36.34	76.50	69.00	7.67	10.00	**81.35**	14.00
FAR	1	0.00	0.00	0.00	0.00	7.43	5.41	0.00	0.00	4.67	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00	7.43	5.41	0.00	0.00	4.67	0.00	0.00	0.00
	Average	0.00	0.00	0.00	0.00	7.43	5.41	0.00	0.00	4.67	0.00	0.00	0.00

while for DyCA, the number of latent variables is also set to 8. The confidence level is set uniformly to 99% in all methods.

The fault detection results for Fault 1 and Fault 2, obtained using various methods, are summarized in Table V. As illustrated in the table, methods that incorporate process dynamics, such as DPCA, DiPCA, SFA, and DyCA, demonstrate superior performance compared to PCA for Fault 1. Notably, SFA and DyCA achieve higher fault detection rates than DPCA and DiPCA by leveraging derivative information. Among the monitoring statistics, the DyCA T^2 statistic achieves the highest FDR for Fault 1 at 90.00%. For Fault 2, although DPCA exhibits a lower FDR than PCA, methods such as DiPCA, DyCA, and SFA outperform both PCA and DPCA. Moreover, DyCA and SFA show superior capabilities in handling process dynamics compared to DiPCA. However, DiPCA demonstrates limitations, as it misclassifies some normal process data. The DyCA T^2 statistic achieves the highest FDR for Fault 2 at 72.70%, while maintaining a satisfactory FAR.

Moreover, as displayed in Table V, the DyCA T^2 statistic demonstrates superior performance over the SPE statistic. This is because DyCA identifies a subspace encapsulating the system's dominant dynamic couplings. Consequently, the T^2 statistic, which monitors this specific subspace, is highly sensitive to faults that disrupt these core dynamic relationships. In contrast, the SPE statistic measures projection errors outside the modeled subspace, which typically correspond to noise or unmodeled dynamics. Given that the BFIP is a tightly coupled system governed by strong physicochemical reactions, significant faults inherently disrupt the dynamic subspace. They are, therefore, more salient in the T^2 statistic and less evident in the residual variations measured by SPE.

To further assess the performance of the proposed methods, the monitoring results for Fault 1 and Fault 2 are presented in Figs. 4 and 5, respectively. Fault 1, referred to as the hanging fault, occurs in the upper region of the blast furnace and leads to a significant reduction in both the quantity and the pressure of the blast. As shown in Fig. 4, DPCA T^2, DiPCA ϕ_v^2, SFA T^2, and DyCA T^2 effectively and promptly detect faulty samples, demonstrating superior performance relative to the T^2 statistic of PCA.

Fault 2, resulting from an insufficient hot air supply to the blast furnace due to a malfunction in the hot blast stove, presents a more challenging detection scenario. Specifically, the fault is caused by a stuck hot blast valve. In contrast to the hanging fault, the detection of this malfunction is more difficult. As illustrated in Fig. 5, the T^2 statistics of SFA and DyCA outperform those of PCA-based methods in detecting the fault. Among all the methods compared, DyCA T^2 delivers the best performance, achieving the highest fault detection rate while maintaining a low false alarm rate.

Table VI lists the offline training and online monitoring times (ms) of various methods for the TEP and BFIP datasets. As summarized in Table VI, although the offline training time of DiPCA (11.4 ms for TEP and 12.5 ms for BFIP) is longer due to iterative auto-covariance maximization and DPCA also incurs additional cost owing to augmented lagged vectors, DyCA and SFA achieve comparable and shorter training times (2.08 ms for TEP and 1.8 ms for BFIP). Moreover, the elapsed time for online monitoring across all methods is less than one second, demonstrating that the proposed DyCA-based fault detection approach can effectively detect abnormalities in real time.

TABLE VI
OFFLINE TRAINING AND ONLINE MONITORING TIMES (MS) OF VARIOUS METHODS FOR THE TEP AND BFIP DATASETS.

	Method	TEP	BFIP
Offline training	PCA	0.617	0.705
	DPCA	4.43	4.46
	DiPCA	11.4	12.5
	SFA	2.59	1.86
	DyCA	2.08	1.80
Online monitoring	PCA	0.058	0.028
	DPCA	0.075	0.033
	DiPCA	0.080	0.039
	SFA	0.067	0.023
	DyCA	0.091	0.026

IV. CONCLUSION

This study proposes a dynamical component extraction based fault detection for IIoT. In the proposed DyCA based approach, DyCA is employed to identify the relevant subspace of multivariate signals. This method enables the decomposition of the process data into deterministic and stochastic components, facilitating a detailed exploration of the temporal structure. The deterministic components (i.e., DyCA components) effectively capture low-dimensional subspaces that represent the system's dynamics while accounting for noise. Based on the extracted deterministic and stochastic components, Hotelling's T^2 and SPE statistics are introduced as monitoring metrics, with their upper control limits determined through kernel density estimation. Experimental evaluations on the widely used Tennessee Eastman process benchmark and a real-world

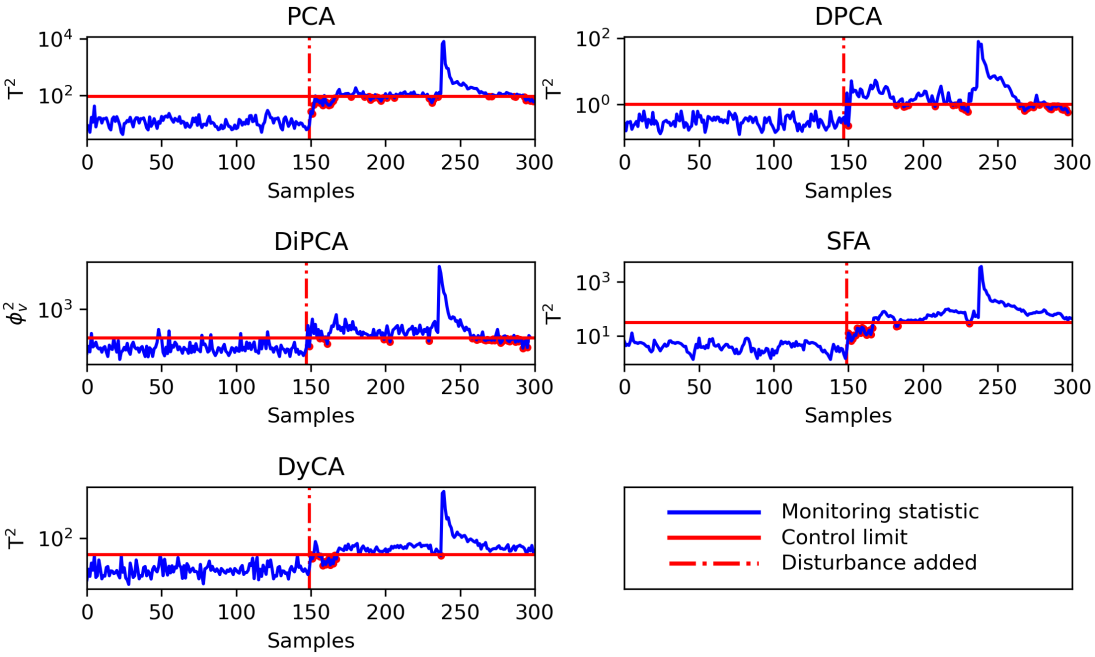

Fig. 4. The monitoring results of PCA T^2, DPCA T^2, DiPCA ϕ_v^2, SFA T^2, DyCA T^2 for fault 1: BFIP.

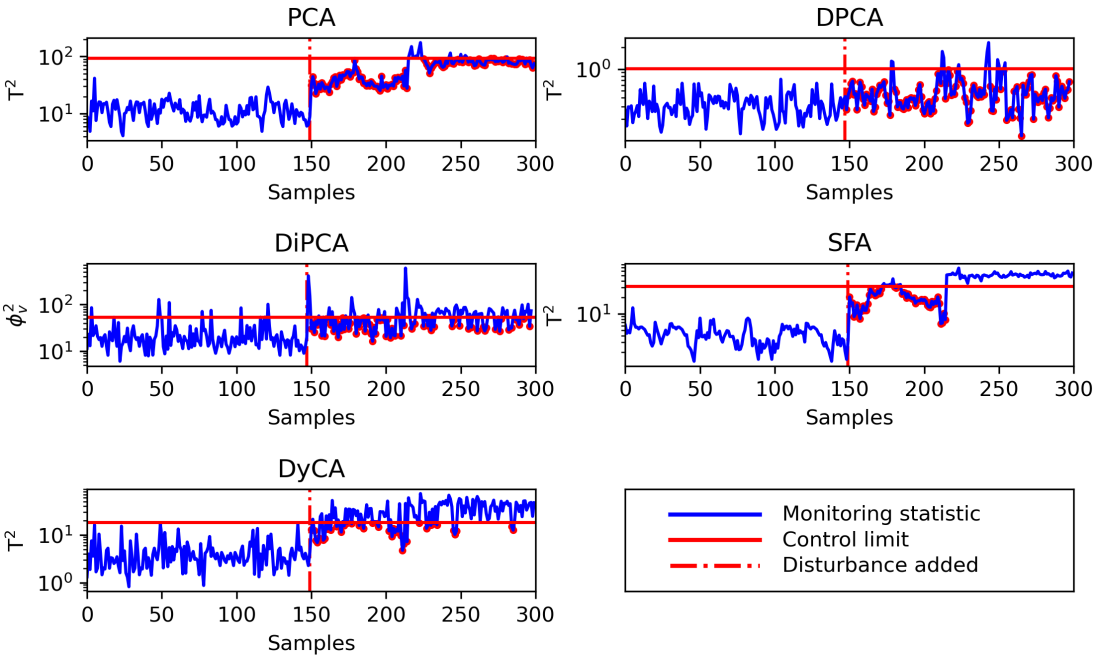

Fig. 5. Monitoring results of PCA T^2, DPCA T^2, DiPCA ϕ_v^2, SFA T^2, DyCA T^2 for fault 2: BFIP.

blast furnace ironmaking process demonstrate the superior performance of the proposed DyCA based fault detection method compared to existing approaches.

Despite its promising performance, the proposed method offers several avenues for future enhancement:

- The current DyCA based fault detection framework is developed under the assumption of linearity. To address the complexities of nonlinear processes, future research will focus on extending the method to account for nonlinear data, thereby improving its applicability and robustness in diverse industrial settings.
- DyCA enables the extraction of ODEs to describe multivariate signals. Further investigation into the extracted ODEs could yield deeper insights and additional information, potentially enhancing the effectiveness and interpretability of fault detection and diagnosis techniques.
- This work focuses primarily on fault detection. A key direction for future research will be the integration of DyCA with causal inference methods, such as Granger causality or transfer entropy for root cause analysis.

REFERENCES

[1] L. Yao and Z. Ge, "Big data quality prediction in the process industry: A distributed parallel modeling framework," *J. Process Control*, vol. 68, pp. 1–13, 2018.

[2] W. He, F. Qian, J. Lam, G. Chen, Q.-L. Han, and J. Kurths, "Quasi-synchronization of heterogeneous dynamic networks via distributed impulsive control: Error estimation, optimization and design," *Automatica*, vol. 62, pp. 249–262, 2015.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Inf.*, vol. 14, no. 11, pp. 4724–4734, 2018.

[4] H. Zhou, C. Yang, and Y. Sun, "Intelligent ironmaking optimization service on a cloud computing platform by digital twin," *Engineering*, vol. 7, no. 9, pp. 1274–1281, 2021.

[5] L. Zhou, Y. Wang, Y. Wu, S. He, and Z. Song, "Quality-relevant modeling and monitoring of industrial cyber-physical systems: The semi-supervised dynamic latent variable models," *IEEE Trans. Ind. Cyber-Phys. Syst.*, vol. 3, pp. 39–47, 2025.

[6] Y. Chi, Y. Dong, Z. J. Wang, F. R. Yu, and V. C. Leung, "Knowledge-based fault diagnosis in industrial internet of things: a survey," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12886–12900, 2022.

[7] Y. Wang, J. Shen, S. Yang, Q. Han, C. Zhao, P. Zhao, and X. Ren, "Knowledge and data dual-driven fault diagnosis in industrial scenarios: A survey," *IEEE Internet Things J.*, 2024.

[8] J. Zheng, L. Zhou, Y. Lyu, Z. Yang, and Z. Ge, "Multi-rate data distillation for deep process monitoring," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–10, 2025.

[9] Y. Chen, H. Bai, S. Li, and X. Zhou, "Dynamic non-gaussian and nonlinear industrial process monitoring using deep analysis of hybrid characteristics," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–12, 2025.

[10] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, 2012.

[11] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, 2015.

[12] Y. Wang, Y. Si, B. Huang, and Z. Lou, "Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017," *Can. J. Chem. Eng.*, vol. 96, no. 10, pp. 2073–2085, 2018.

[13] Z. Ge and Z. Song, *Multivariate statistical process control: Process monitoring methods and applications*. Springer Science & Business Media, 2012.

[14] H. Ma, Y. Wang, X. Liu, J. Yuan, and Z. Ji, "Quality-oriented dynamic sparse latent variable detection approach for industrial iot based on joint spatial and temporal decomposition," *IEEE Internet Things J.*, 2025.

[15] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nat. Rev. Methods Primers*, vol. 2, no. 1, p. 100, 2022.

[16] B. M. Wise, N. Ricker, D. Veltkamp, and B. R. Kowalski, "A theoretical basis for the use of principal component models for monitoring multivariate processes," *Process Control Qual.*, vol. 1, no. 1, pp. 41–51, 1990.

[17] J. Zheng, Z. Yang, and Z. Ge, "Deep residual principal component analysis as feature engineering for industrial data analytics," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–10, 2024.

[18] H. Fan, X. Lai, S. Du, W. Yu, C. Lu, and M. Wu, "Distributed monitoring with integrated probability pca and mrmr for drilling processes," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[19] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 179–196, 1995.

[20] J. Huang and X. Yan, "Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and bayesian inference," *Chemom. Intell. Lab. Syst.*, vol. 148, pp. 115–127, 2015.

[21] A. H. de Andrade Melani, M. A. de Carvalho Michalski, R. F. da Silva, and G. F. M. de Souza, "A framework to automate fault detection and diagnosis based on moving window principal component analysis and bayesian network," *Reliab. Eng. Syst. Saf.*, vol. 215, p. 107837, 2021.

[22] G. Li, S. J. Qin, and D. Zhou, "A new method of dynamic latent-variable modeling for process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6438–6445, 2014.

[23] L. Zhou, G. Li, Z. Song, and S. J. Qin, "Autoregressive dynamic latent variable models for process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 366–373, 2016.

[24] 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.

[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] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 436–450, 2012.

[27] B. Du, L. Ru, C. Wu, and L. Zhang, "Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9976–9992, 2019.

[28] C. Shang, F. Yang, X. Gao, X. Huang, J. A. Suykens, and D. Huang, "Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis," *AlChE J.*, vol. 61, no. 11, pp. 3666–3682, 2015.

[29] S. Zhang and C. Zhao, "Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3773–3783, 2018.

[30] X. Ma, Y. Si, Z. Yuan, Y. Qin, and Y. Wang, "Multistep dynamic slow feature analysis for industrial process monitoring," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9535–9548, 2020.

[31] M. V. Rishi and A. K. Tangirala, "Probabilistic adaptive slow feature analysis for state estimation and classification," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–15, 2024.

[32] C. Uhl, M. Kern, M. Warmuth, and B. Seifert, "Subspace detection and blind source separation of multivariate signals by dynamical component analysis (dyca)," *IEEE Open J. Signal Process.*, vol. 1, pp. 230–241, 2020.

[33] P.-E. P. Odiowei and Y. Cao, "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations," *IEEE Trans. Ind. Inf.*, vol. 6, no. 1, pp. 36–45, 2009.

[34] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, 1993.

[35] S. Puttinger and H. Stocker, "Toward a better understanding of blast furnace raceway blockages," *Steel Res. Int.*, vol. 91, no. 12, p. 2000227, 2020.