SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2026, Vol. 69, Iss. 7, 172208:1–172208:19
https://doi.org/10.1007/s11432-025-4891-0

A novel quality-related dynamic slow feature analysis method for process monitoring

Chen ZHANG, Xiangyu KONG*, Meizhi LIU, Changhua HU & Lixin WANG

Laboratory of Intelligent Control, Rocket Force University of Engineering, Xi'an 710025, China

Received 7 April 2025/Revised 28 September 2025/Accepted 21 February 2026/Published online 8 June 2026

Abstract In recent decades, multivariate statistical process monitoring (MSPM) has established itself as a dominant technique in process monitoring and fault detection. However, traditional MSPM methods exhibit suboptimal performance in quality-related dynamic process monitoring, as they lack the capability to elucidate whether faults occurring in process monitoring correlate with quality variables or not. To solve the problem, a novel quality-related dynamic slow feature analysis (QRDSFA) method is proposed, where a new objective function is established based on dynamic partial least squares (DPLS) and SFA. Afterward, a combined method incorporating the Lagrange multipliers and a self-search strategy is introduced to solve the multi-objective optimization problems. Furthermore, a vector autoregression (VAR) model is introduced to detect anomalies in the dynamic process at the current moment. Finally, statistical indicators are developed using latent variables and residuals to track dynamic changes pertinent to quality variables. Comprehensive case studies involving numerical simulations, the Tennessee Eastman process, and electric servo systems demonstrate that the proposed methodology exhibits superior monitoring performance compared to other dynamic approaches.

Keywords quality-related, slow feature analysis, dynamic features, process monitoring, self-search strategy

Citation Zhang C, Kong X Y, Liu M Z, et al. A novel quality-related dynamic slow feature analysis method for process monitoring. Sci China Inf Sci, 2026, 69(7): 172208, https://doi.org/10.1007/s11432-025-4891-0

1 Introduction

Data-driven process monitoring is a critical task for complex system health management and a key technology for ensuring the reliable operation of large equipment. Multivariate statistical process monitoring (MSPM) [1–3], encompassing techniques like principal component analysis (PCA) [4–6], partial least squares (PLS) [7–9], and independent component analysis (ICA) [10–12], is a widely adopted data-driven approach for process monitoring and fault detection. By projecting high-dimensional process data onto a low-dimensional subspace, traditional MSPM methods construct monitoring statistics and achieve process monitoring.

To achieve precise control and optimization of complex systems or equipment operations, process monitoring methods need to screen out quality-related variables and extract the dynamic characteristics of time series data. Firstly, in quality-related process monitoring, process variables refer to variables that can be directly observed during the processes, such as voltage, current, and pressure; while quality-related variables stand for variables related to equipment performance or production quality indicators, such as the nozzle swing angle of rocket launch systems, feedback potential of servo mechanisms and product concentration in chemical processes [13]. Secondly, process data typically exhibit a dynamic relationship due to the influence of feedback control systems, random noise, and process interference. In other words, there is a correlation between the data samples obtained at a certain moment and the data samples before and after sampling. However, in quality-related dynamic process monitoring, traditional MSPM methods have two limitations. On the one hand, despite that traditional MSPM methods can effectively identify changes in process variables, they cannot reveal whether faults occurring in process variables are related to output quality variables or not, thus leading to false alarms of quality-unrelated faults [14]. On the other hand, traditional MSPM methods assume that process data are sampled independently, which ignores the dynamic relationships commonly present in process data, making it difficult to effectively monitor the health status of dynamic systems [15].

For quality-related process monitoring issues, PLS is proposed to extract the relevant features of process variables and quality variables. Unlike the full index process monitoring method, PLS creates a link from process variables

* Corresponding author (email: xiangyukong01@163.com)

to quality variables. By monitoring changes in process variables and detecting abnormal states of quality variables in advance, PLS is able to achieve early warning of key performance indicators and has been widely applied [16]. Chen et al. [17] proposed an improved deep kernel PLS model using the well-designed reinforced learning strategy, which improves the prediction performance with nearly no extra training costs. Yu et al. [18] proposed a new PLS regression trees modeling algorithm when there are multicollinear relationships between predictor variables. Chu et al. [19] proposed a method for operating performance assessment on the basis of kernel locally linear embedding PLS, whose effectiveness is verified by two cases. Although PLS has been widely used, its application is based on the assumption that there is a static relationship between process variables and quality variables. However, in actual processes, process variables and quality variables tend to rely on past data and exhibit dynamic changes. Therefore, most PLS-based models cannot meet the monitoring performance requirements of dynamic processes well [20].

For dynamic process monitoring issues, a large number of algorithms have been devised recently. For example, dynamic PCA (DPCA), building on standard PCA, has been proposed to facilitate dynamic process monitoring [21]. By utilizing delayed measurement values, DPCA constructs an augmented data matrix and performs feature extraction to capture both static and dynamic relationships in the mining process. Inspired by DPCA, dynamic PLS (DPLS) [22] also achieves quality-related dynamic process monitoring by constructing an augmented data matrix, such as DMPLS [23] and DTPLS [24]. However, these methods have several common drawbacks. First of all, traditional MSPM algorithms mix static and dynamic information in the augmented matrix, which may result in an ambiguous representation of the dynamic structure of the underlying process. Secondly, as the number of time delays increases, the dimensionality of the augmented data matrix sharply increases, thereby increasing the computational complexity [25].

To extract important dynamic information from process data, slow feature analysis (SFA) has been proposed and applied [26]. SFA is an unsupervised method used to extract features from time series datasets that change slowly over time. The fundamental principle of SFA is that latent variables that change slowly over time carry crucial information about the data, whereas latent variables that change rapidly are usually linked to noise. In addition, the successful application of SFA in various fields also confirms this principle. He et al. [27] proposed a multivariate probabilistic SFA model to monitor the behavior of nonstationary processes, which showed obvious superiority in prediction precision and real value tracing. Zhang et al. [28] introduced a novel continual learning-based probabilistic SFA algorithm, which achieved significant performance in actual industrial processes. Shang et al. [29] put forward a recursive SFA algorithm for adaptive process monitoring, which could update model parameters when new samples arrive. However, given that SFA is an unsupervised method, features extracted by SFA from input variables are not correlated with quality variables, thus unable to reflect the health status of quality variables.

In summary, traditional MSPM methods are not ideal for quality-related process monitoring involving dynamic data. Therefore, one feasible approach is to combine PLS and SFA to extract quality-related dynamic features. Chiplunkar et al. [30] proposed a combination of PLS quality-related feature extraction and SFA time-delay feature extraction. Besides, an objective function similar to classical PLS was established, and the proposed method was applied to three case studies. Zhu et al. [31] improved the method and verified the effectiveness of PLS-SFA through mathematical calculations and ethylene process examples. The research results provided strong support for improving energy efficiency and formulating production strategies for high energy-consuming factories. Although the above method successfully extracts quality-related dynamic features from process variables, it can only extract first-order dynamic features of process variables and cannot provide a comprehensive characterization of process dynamic characteristics.

In order to extract comprehensive quality-related dynamic process characteristics, this article proposes a novel quality-related dynamic slow feature analysis (QRDSFA) method. Firstly, on the basis of DPLS and SFA, a new objective function is established to extract comprehensive quality-related dynamic characteristics. Afterward, a new regression algorithm is proposed, which uses the Lagrange multipliers to solve the objective function and determines parameters through a self-search strategy. Furthermore, a vector autoregression (VAR) model is introduced to detect anomalies in the dynamic process at the current moment. Finally, statistical indicators are developed using latent variables and residuals to track dynamic changes pertinent to quality variables. Compared with [30,31], the proposed QRDSFA method has two unique advancements. On the one hand, the VAR model is used for current-moment anomaly detection. VAR models excel by integrating multiple explanatory variables to model complex latent relationships and capturing both short- and long-term interdependencies among variables, thereby significantly enhancing fitting accuracy. On the other hand, a self-search strategy is used for parameter optimization. The self-search strategy integrated with Lagrange multipliers effectively transforms constrained multi-objective problems into unconstrained ones, enhances global search capability to escape local optima, and demonstrates strong robustness in complex high-dimensional optimization scenarios. The main contributions are as follows.

(1) A novel QRDSFA method is proposed to extract comprehensive quality-related dynamic process characteristics.

(2) A hybrid approach that integrates Lagrange multipliers with a self-search strategy is introduced for solving multi-objective optimization problems.

(3) A VAR model is introduced to calculate the features at the current time rather than features mixed with noise from other times.

The rest of the article is organized as follows. In Section 2, the related studies are reviewed in detail. In Section 3, the idea of the proposed method and its implementing procedures for offline modeling and online monitoring are described. Section 4 validates the effectiveness of the proposed method on a numerical simulation case, the Tennessee Eastman process and the electric servo mechanisms. In Section 5, some discussions of the proposed algorithm are provided. Finally, the conclusion is summarized in Section 6.

2 Preliminaries

2.1 DPLS model

Given input matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ of n samples and m process variables, and output matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$ of n samples and p observation variables. The traditional PLS model is expressed as follows:

$$\begin{cases} \boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \tilde{\boldsymbol{X}}, \\ \boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}} + \tilde{\boldsymbol{Y}}, \end{cases} \tag{1}$$

where $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_K]$ and $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K]$ are the score matrix and loading matrix of $\boldsymbol{X}$, respectively, while $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K]$ and $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_K]$ represent the score matrix and loading matrices of $\boldsymbol{Y}$, respectively. The modeling residuals of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are expressed by $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$, while K represents the number of latent variables.

To better align the objective function with dynamic scenarios, the subsequent objective function is formulated for each iteration of the outer modeling process within this study

$$\begin{aligned} &\max\ (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} \\ &\text{s.t. } \|\boldsymbol{w}\| = \|\boldsymbol{\beta}\| = \|\boldsymbol{c}\| = 1, \end{aligned} \tag{2}$$

where $(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}}) = [\boldsymbol{\beta}_{(0)}\boldsymbol{w}^{\mathrm{T}}, \boldsymbol{\beta}_{(1)}\boldsymbol{w}^{\mathrm{T}}, \ldots, \boldsymbol{\beta}_{(q-1)}\boldsymbol{w}^{\mathrm{T}}]$ is the Kronecker product, $\boldsymbol{X}_{(i)}$ represents input data with i time lags, q is the number of lagged time, and $\boldsymbol{\beta}_{(i)}$ is a weight coefficient for $\boldsymbol{X}_{(i)}w$. By identifying an optimal direction vector $\boldsymbol{w}$ and a coefficient vector $\boldsymbol{\beta} = [\boldsymbol{\beta}_{(0)}, \ldots, \boldsymbol{\beta}_{(q-1)}]^{\mathrm{T}}$, the dynamic linear relation of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is maximized.

To refine the expression more effectively, let $\boldsymbol{X}_d = [\boldsymbol{X}_{(0)}, \boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(q-1)}]$. Afterward, the DPLS model is as follows:

$$\begin{cases} \boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \tilde{\boldsymbol{X}}, \\ \boldsymbol{Y} = \boldsymbol{H}_1(\boldsymbol{z}^{-1})\boldsymbol{t}_1\boldsymbol{q}_1^{\mathrm{T}} + \cdots + \boldsymbol{H}_K(\boldsymbol{z}^{-1})\boldsymbol{t}_K\boldsymbol{q}_K^{\mathrm{T}} + \tilde{\boldsymbol{Y}}. \end{cases} \tag{3}$$

Consequently, the aforementioned algorithm will not increase the dimension of $\boldsymbol{w}$, but delineates the dynamic correlation between $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Moreover, unlike conventional approaches relying on augmented data matrices, the DPLS model adopted here incorporates time-dependent dynamics directly into the internal structure of PLS. By establishing dynamic relationships between input and output scores, rather than static mappings, this method significantly enhances the monitoring of quality-relevant anomalies.

2.2 SFA model

Assuming an input signal $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \ldots, \boldsymbol{x}_n(t)]$ and a nonlinear function $\boldsymbol{g}_j(\cdot)_{j=1}^n$ with $\boldsymbol{s}_j(t) = \boldsymbol{g}_j(x(t))$ so that for every $j = 1, 2, \ldots, n$,

$$\min\ \left\langle \dot{\boldsymbol{s}}_j^2 \right\rangle_t, \tag{4}$$

under the constraints

$$\left\langle \boldsymbol{s}_j \right\rangle_t = 0, \tag{5}$$

$$\left\langle \boldsymbol{s}_j^2 \right\rangle_t = 1, \tag{6}$$

$$\forall i \neq j : \langle \boldsymbol{s}_i \boldsymbol{s}_j \rangle_t = 0, \tag{7}$$

where $\dot{\boldsymbol{s}}_j$ is the first-order derivative or temporal variation of the output signal $\boldsymbol{s}_j$ and $\langle \cdot \rangle_t$ denotes temporal averaging as follows:

$$\langle \boldsymbol{y} \rangle_t = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} \boldsymbol{y}(t) dt. \tag{8}$$

Afterward, the mapping of SFA is as follows:

$$\boldsymbol{s}(t) = \boldsymbol{w}\boldsymbol{x}(t), \tag{9}$$

where $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n]$ is the coefficient matrix for SFA and $\boldsymbol{s}(t) = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n]^{\mathrm{T}}$ is the slow feature. Considering that Eqs. (4) and (7) can be modified as follows, respectively:

$$\left\langle \dot{\boldsymbol{s}}_j^2 \right\rangle_t = \left\langle \left(\boldsymbol{w}_j^{\mathrm{T}} \dot{\boldsymbol{x}} \right)^2 \right\rangle_t = \boldsymbol{w}_j^{\mathrm{T}} \left\langle \dot{\boldsymbol{x}} \dot{\boldsymbol{x}}^{\mathrm{T}} \right\rangle_t \boldsymbol{w}_j = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}_j, \tag{10}$$

$$\left\langle \boldsymbol{s}_j^2 \right\rangle_t = \left\langle \left(\boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x} \right)^2 \right\rangle_t = \boldsymbol{w}_j^{\mathrm{T}} \left\langle \boldsymbol{x} \boldsymbol{x}^{\mathrm{T}} \right\rangle_t \boldsymbol{w}_j = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{B} \boldsymbol{w}_j, \tag{11}$$

where $\boldsymbol{A} = \left\langle \dot{\boldsymbol{x}} \dot{\boldsymbol{x}}^{\mathrm{T}} \right\rangle_t$, $\boldsymbol{B} = \left\langle \boldsymbol{x} \boldsymbol{x}^{\mathrm{T}} \right\rangle_t$.

Therefore, the objective function of SFA is rewritten as follows:

$$\begin{aligned} &\min \ \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}_j \\ &\text{s.t.} \ \ \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{B} \boldsymbol{w}_j = 0 \ \forall i \neq j. \end{aligned} \tag{12}$$

As shown in (12), SFA is an unsupervised method where features extracted from input variables may not have a good correlation with output quality, resulting in poor process monitoring results.

3 Methodology

In this section, a novel QRDSFA model is established, which combines DPLS and SFA to extract complete quality-related dynamic characteristics. Based on the QRDSFA model, a novel quality-related dynamic process monitoring strategy is also designed.

3.1 Objective function construction

Given input matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ of n samples and m process variables, and output matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times p}$ of n samples and p observation variables. As discussed in the previous section, to extract complete quality-related dynamic characteristics, Eqs. (2) and (12) must be satisfied simultaneously, which is shown as follows:

$$\begin{aligned} &\max \ (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}}) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{c} \\ &\min \ \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}_j \\ &\text{s.t.} \ \ \|\boldsymbol{w}\| = \|\boldsymbol{\beta}\| = \|\boldsymbol{c}\| = 1, \end{aligned} \tag{13}$$

where $\boldsymbol{A} = \left\langle \dot{\boldsymbol{X}} \dot{\boldsymbol{X}}^{\mathrm{T}} \right\rangle_t$.

Obviously, solving (13) is a multi-objective optimization problem. However, no solitary solution can be identified that optimizes all objectives concurrently. For the multi-objective problem, one of the most commonly used solutions is to linearly scalarize the multi-objective function, which combines all objective functions into one function [32]. Therefore, the linear scalarization of (13) can be formulated as follows:

$$\begin{aligned} &\max \ (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}}) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{c} - \alpha \ \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}_j \\ &\text{s.t.} \ \ \|\boldsymbol{w}\| = \|\boldsymbol{\beta}\| = \|\boldsymbol{c}\| = 1. \end{aligned} \tag{14}$$

The weight of objective ($\alpha \geqslant 0$) is the parameter of the scalarization and the procedure of tuning α is presented in the next subsection. Therefore, the objective function of the QRDSFA model is established as (14).

For optimization problems subject to equality constraints, the application of Lagrange multipliers offers an effective framework by converting the constrained formulation into an equivalent unconstrained problem, thereby streamlining the analytical derivation and numerical resolution of the optimization model [33]. It allows us to

efficiently incorporate constraints into the objective function, simplifying the problem and facilitating analytical solutions. Alternative methods may not offer such a straightforward or elegant way to address the specific constraint structure in our case. Therefore, the Lagrange multipliers method is introduced to solve the optimization model (14), and the Lagrange function is constructed as follows:

$$\begin{aligned}\max\ \boldsymbol{J} = &(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} - \alpha \boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} \\ &+ \frac{1}{2}\lambda_w(1-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}) + \frac{1}{2}\lambda_c(1-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}) + \frac{1}{2}\lambda_\beta(1-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}).\end{aligned} \tag{15}$$

Taking derivatives with respect to $\boldsymbol{w}$, $\boldsymbol{c}$ and $\boldsymbol{\beta}$ leads to

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\beta}} = (\boldsymbol{I}_q \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} - \lambda_\beta \boldsymbol{\beta} = 0, \tag{16}$$

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{w}} = (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m)\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} - \lambda_w \boldsymbol{w} - \alpha(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w} = 0, \tag{17}$$

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{c}} = [(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}]^{\mathrm{T}} - \lambda_c \boldsymbol{c} = 0. \tag{18}$$

Substituting (18) into (16) and (17) respectively, we have

$$(\boldsymbol{I}_q \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta} \otimes \boldsymbol{w}) = \lambda_\beta \lambda_c \boldsymbol{\beta}, \tag{19}$$

$$(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m)\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta} \otimes \boldsymbol{w}) = \lambda_c(\lambda_w \boldsymbol{I}_m + \alpha(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}))\boldsymbol{w}. \tag{20}$$

Considering that

$$\boldsymbol{\beta} \otimes \boldsymbol{w} = (\boldsymbol{I}_q \otimes \boldsymbol{w})\boldsymbol{\beta} = (\boldsymbol{\beta} \otimes \boldsymbol{I}_m). \tag{21}$$

We can define

$$\begin{cases} \boldsymbol{S} \equiv \boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d \in \mathbb{R}^{mq \times mq}, \\ \boldsymbol{S}_{\boldsymbol{w}} \equiv (\boldsymbol{I}_q \otimes \boldsymbol{w})^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{I}_q \otimes \boldsymbol{w}) \in \mathbb{R}^{q \times q}, \\ \boldsymbol{S}_{\boldsymbol{\beta}} \equiv (\boldsymbol{\beta} \otimes \boldsymbol{I}_m)^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{\beta} \otimes \boldsymbol{I}_m) \in \mathbb{R}^{m \times m}. \end{cases} \tag{22}$$

Substituting (22) into (19) and (20) respectively, we can obtain

$$\begin{cases} \boldsymbol{S}_{\boldsymbol{w}}\boldsymbol{\beta} = \lambda_\beta \lambda_c \boldsymbol{\beta}, \\ \boldsymbol{S}_{\boldsymbol{\beta}}\boldsymbol{w} = \lambda_c(\lambda_w \boldsymbol{I}_m + \alpha(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}))\boldsymbol{w}. \end{cases} \tag{23}$$

As shown in (23), $\boldsymbol{\beta}$ is the eigenvector of matrix $\boldsymbol{S}_{\boldsymbol{w}}$, while $\boldsymbol{w}$ is the generalized eigenvector of matrix $\boldsymbol{S}_{\boldsymbol{\beta}}$ and $(\lambda_w \boldsymbol{I}_m + \alpha(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}))$.

In addition, from (18) we can obtain

$$\boldsymbol{c} = \frac{1}{\lambda_c}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta} \otimes \boldsymbol{w}). \tag{24}$$

Substituting (24) into (14), we can obtain

$$\begin{aligned}\boldsymbol{J}_{\max} &= (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} - \alpha \boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} \\ &= \frac{1}{\lambda_c}(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta} \otimes \boldsymbol{w}) - \alpha \boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}.\end{aligned} \tag{25}$$

Substituting (21) and (19) into (25), we can obtain

$$\begin{aligned}\boldsymbol{J}_{\max} &= \frac{1}{\lambda_c}(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta} \otimes \boldsymbol{w}) - \alpha \boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} \\ &= \frac{1}{\lambda_c}\boldsymbol{\beta}^{\mathrm{T}}\lambda_\beta \lambda_c \boldsymbol{\beta} - \alpha \boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} \\ &= \lambda_\beta \boldsymbol{I}_m - \alpha \boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}.\end{aligned} \tag{26}$$

Substituting (21) and (20) into (25), we can obtain

$$\begin{aligned}\boldsymbol{J}_{\max} &= \frac{1}{\lambda_c}\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\beta}\otimes\boldsymbol{I}_m)^{\mathrm{T}}\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta}\otimes\boldsymbol{w}) - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}\\ &= \frac{1}{\lambda_c}\boldsymbol{w}^{\mathrm{T}}\lambda_c(\lambda_w\boldsymbol{I}_m + \alpha(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}))\boldsymbol{w} - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}\\ &= \lambda_w\boldsymbol{I}_m + \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}.\end{aligned} \tag{27}$$

In addition, substituting (16) into (18), and (17) into (18), respectively, we can obtain

$$\begin{cases}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{I}_q\otimes\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} = \lambda_\beta\lambda_c\boldsymbol{c},\\ \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}\otimes\boldsymbol{I}_m)\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} = (\lambda_w\boldsymbol{I}_m + \alpha(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}))\lambda_c\boldsymbol{c}.\end{cases} \tag{28}$$

As shown in (28), vector $\boldsymbol{c}$ is the eigenvector of the above matrix.
From (24) we can obtain

$$(\boldsymbol{\beta}^{\mathrm{T}}\otimes\boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y} = \lambda_c\boldsymbol{c}^{\mathrm{T}}. \tag{29}$$

Substituting (29) into (14), we can obtain

$$\begin{aligned}\boldsymbol{J}_{\max} &= (\boldsymbol{\beta}^{\mathrm{T}}\otimes\boldsymbol{w}^{\mathrm{T}})\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{c} - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}\\ &= \lambda_c\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}\\ &= \lambda_c\boldsymbol{I}_m - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}.\end{aligned} \tag{30}$$

As shown in (26), (27) and (30), the solution of the objective function satisfies the following relationship:

$$\begin{aligned}\boldsymbol{J}_{\max} &= \lambda_w\boldsymbol{I}_m + \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}\\ &= \lambda_\beta\boldsymbol{I}_m - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}\\ &= \lambda_c\boldsymbol{I}_m - \alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}.\end{aligned} \tag{31}$$

However, as shown in (22), $\boldsymbol{S}_{\boldsymbol{w}}$ is related to $\boldsymbol{w}$ and $\boldsymbol{S}_{\boldsymbol{\beta}}$ is related to $\boldsymbol{\beta}$. Therefore, it is difficult to determine vector $(\boldsymbol{w},\boldsymbol{\beta})$ simply by (23). A more structured approach involves a two-phase optimization strategy: initially establishing the model weights $\boldsymbol{w}$, followed by a systematic search for $(\boldsymbol{w},\boldsymbol{\beta})$ by (23), which is shown in Algorithm 1.

After calculating $\boldsymbol{w}$, $\boldsymbol{\beta}$ and $\boldsymbol{c}$, the QRDSFA overall modeling is constructed as follows:

$$\begin{cases}\boldsymbol{t} = \boldsymbol{X}\boldsymbol{w},\\ \boldsymbol{t}_d = \boldsymbol{X}_d(\boldsymbol{\beta}\otimes\boldsymbol{w}),\\ \boldsymbol{q} = (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{t}_d)/(\boldsymbol{t}_d^{\mathrm{T}}\boldsymbol{t}_d),\\ \boldsymbol{p} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{t})/(\boldsymbol{t}^{\mathrm{T}}\boldsymbol{t}).\end{cases} \tag{32}$$

3.2 Parameter determination

In this subsection, two important parameters are discussed and determined: the number of latent variables K and the weight of objective α.

(1) Parameter determination of K. Currently, there exists no definitive methodology for determining K, thus cross-validation is commonly employed as a means to estimate it. However, considering that latent variables require extracting slow features from process variables, a more explicit standard [33] of determining K is introduced as follows.

On the one hand, too few slow features may not fully reflect the characteristics of the process; on the other hand, excessive low-frequency features can introduce noise, reducing both monitoring effectiveness and detection accuracy. Given that the slower the variables vary, the more intrinsic information they contain, we propose eliminating slow features whose temporal dynamics exceed those of all input variables, with their total count defined as follows:

$$K_e = \mathrm{card}\left\{s_i|\Delta(s_i) > \max_j^q\{\Delta(x_j)\}\right\}, \tag{35}$$

Algorithm 1 QRDSFA.

Step 1. Set $\boldsymbol{w}_i = [1, 0, \ldots, 0]^{\mathrm{T}}$, $\lambda_w = 0.1$. Calculate $\boldsymbol{S}_{\boldsymbol{w}}$ by (22) and solve the following problem:

$$\boldsymbol{S}_{\boldsymbol{w}}\boldsymbol{\beta}_i = \lambda_\beta \lambda_c \boldsymbol{\beta}_i, \tag{33}$$

where $\boldsymbol{\beta}_i$ is the eigenvector associated with the maximum eigenvalue.
Step 2. Calculate α by Algorithm 2, which is elaborated in detail in the next subsection.
Step 3. Calculate $\boldsymbol{S}_\beta$ by (22) and solve the following generalized eigenvector solving problem:

$$\boldsymbol{S}_\beta \boldsymbol{w}_i = \lambda_c(\lambda_w \boldsymbol{I}_m + \alpha(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}))\boldsymbol{w}_i, \tag{34}$$

where $\boldsymbol{w}_i$ is the generalized eigenvector associated with the maximum eigenvalue, λ_c is the generalized eigenvalue.
Step 4. Calculate $\boldsymbol{c}$ by (24).
Step 5. Calculate λ_w by (17).
Step 6. Iterate steps 1–5 until $\boldsymbol{w}_i$ and $\boldsymbol{\beta}_i$ converge.
Step 7. $\boldsymbol{t}_i = \boldsymbol{X}_i \boldsymbol{w}_i$, $\boldsymbol{t}_{di} = \sum_{j=0}^{q-1} \boldsymbol{\beta}_{i(j)} \boldsymbol{t}_{i(j)}$, $\boldsymbol{q}_i = \boldsymbol{Y}_i^{\mathrm{T}} \boldsymbol{t}_{di} / \boldsymbol{t}_{di}^{\mathrm{T}} \boldsymbol{t}_{di}$, $\boldsymbol{p}_i = \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{t}_i / \boldsymbol{t}_i^{\mathrm{T}} \boldsymbol{t}_i$.
Step 8. $\boldsymbol{X}_{i+1} = \boldsymbol{X}_i - \boldsymbol{t}_i \boldsymbol{p}_i^{\mathrm{T}}$, $\boldsymbol{X}_{d,\,i+1} = [\boldsymbol{X}_{i+1(0)}, \boldsymbol{X}_{i+1(1)}, \ldots, \boldsymbol{X}_{i+1(q-1)}]$, $\boldsymbol{Y}_{i+1} = \boldsymbol{Y}_i - \boldsymbol{t}_{di} \boldsymbol{q}_i^{\mathrm{T}}$, $i = i + 1$, return to step 1 until $i > K$.

where card $\{\}$ denotes the number of elements in a certain set, and $\max_j^q \{\Delta(x_j)\}$ stands for the q-upper quantile of the set $\{\Delta(x_j)\}$. Usually, we can set $q = 0.1$.

Then, the number of dominant variables K is calculated as follows:

$$K = m - K_e. \tag{36}$$

(2) Parameter determination of α. As shown in (14), the objective function is shown as follows:

$$\max\ \boldsymbol{J} = (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{w}^{\mathrm{T}}) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{c} - \alpha \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}. \tag{37}$$

Substituting (21) into (20), we can obtain

$$(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}_d (\boldsymbol{\beta} \otimes \boldsymbol{I}_m) \boldsymbol{w} = \lambda_c(\lambda_w \boldsymbol{I}_m + \alpha(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}))\boldsymbol{w}. \tag{38}$$

In other words, we can reach the following result:

$$\boldsymbol{w} \propto (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}_d (\boldsymbol{\beta} \otimes \boldsymbol{I}_m). \tag{39}$$

Combined with (37) and (39), a new optimization objective function for solving α is proposed as follows:

$$\max\ \boldsymbol{J}_{new} = \boldsymbol{w}^{\mathrm{T}} (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}_d (\boldsymbol{\beta} \otimes \boldsymbol{I}_m) \boldsymbol{w} - \alpha \boldsymbol{w}^{\mathrm{T}} \left\langle \dot{\boldsymbol{X}} \dot{\boldsymbol{X}}^{\mathrm{T}} \right\rangle_t \boldsymbol{w}. \tag{40}$$

By maximizing squared covariance between $\boldsymbol{X}_d$ and $\boldsymbol{Y}$, the extracted features will exhibit mutual correlation [30]. Therefore, Eq. (40) can be rewritten as follows:

$$\max\ \boldsymbol{J}_{new} = \boldsymbol{w}^{\mathrm{T}} \Big(\big(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m\big) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}_d \big(\boldsymbol{\beta} \otimes \boldsymbol{I}_m\big) - \alpha \left\langle \dot{\boldsymbol{X}} \dot{\boldsymbol{X}}^{\mathrm{T}} \right\rangle_t \Big) \boldsymbol{w}. \tag{41}$$

As shown in (41), the value of the new objective function is mainly influenced by two parts. The first part is $(\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}_d (\boldsymbol{\beta} \otimes \boldsymbol{I}_m)$, which represents the covariance matrix between dynamic input data and output data. The other part is $\left\langle \dot{\boldsymbol{X}} \dot{\boldsymbol{X}}^{\mathrm{T}} \right\rangle_t$, which represents the covariance matrix of input data time variation. The objective function's value is clearly proportional to the difference between the two matrix parts with weight α. From the form of the two matrices, it can be analyzed that both of the two matrices are positive definite matrices. However, calculated by taking a difference, $\boldsymbol{J}_{new}$ is not always positive definite. As α increases beyond a critical threshold, the matrix loses its positive definiteness, transitioning through an indefinite phase and potentially progressing to negative semi-definite or even negative definite states as the parameter continues to grow [30].

In other words, a regression model is shown as follows:

$$\tilde{\boldsymbol{J}} = (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m) \boldsymbol{X}_d^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}_d (\boldsymbol{\beta} \otimes \boldsymbol{I}_m) - \alpha \left\langle \dot{\boldsymbol{X}} \dot{\boldsymbol{X}}^{\mathrm{T}} \right\rangle_t. \tag{42}$$

When α exceeds a certain threshold, $\tilde{\boldsymbol{J}}$ begins to oscillate, in which the highest eigenvalue of $\tilde{\boldsymbol{J}}$ turns to zero. Therefore, we traverse the value of α until the critical value appears, where $\tilde{\boldsymbol{J}}$ becomes unstable. Based on the aforementioned analysis, a self-search strategy is proposed for selecting α, which is shown in Algorithm 2.

Algorithm 2 Self-search strategy.

Step 1. Initialization by setting $\alpha = 1$ and $\Delta = 0$.
Step 2. **While** $\Delta < 0$
Step 3. $\alpha = \alpha + 1$;
Step 4. Calculate $\tilde{\boldsymbol{J}}$ as follows:

$$\tilde{\boldsymbol{J}} = (\boldsymbol{\beta}^{\mathrm{T}} \otimes \boldsymbol{I}_m)\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}_d(\boldsymbol{\beta} \otimes \boldsymbol{I}_m) - \alpha\left\langle \dot{\boldsymbol{X}}\dot{\boldsymbol{X}}^{\mathrm{T}}\right\rangle_t; \tag{43}$$

Step 5. Perform matrix decomposition on matrix $\tilde{\boldsymbol{J}}$;
Step 6. Find the maximum eigenvalue of $\tilde{\boldsymbol{J}}$ and assign it to Δ;
Step 7. **Endwhile**

3.3 Quality-related process monitoring

On the basis of the QRDSFA model, input data $\boldsymbol{X}$ and output data $\boldsymbol{Y}$ are dynamically modeled as follows:

$$\begin{cases} \boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \tilde{\boldsymbol{X}}, \\ \boldsymbol{Y} = \boldsymbol{H}_1(z^{-1})\boldsymbol{t}_1\boldsymbol{q}_1^{\mathrm{T}} + \cdots + \boldsymbol{H}_K(z^{-1})\boldsymbol{t}_K\boldsymbol{q}_K^{\mathrm{T}} + \tilde{\boldsymbol{Y}}, \end{cases} \tag{44}$$

where $\boldsymbol{H}_i(z^{-1}) = \boldsymbol{\beta}_i^{\mathrm{T}}\boldsymbol{z}$ stands for the dynamic relation between output data $\boldsymbol{Y}$ and dynamic score $\boldsymbol{t}_i$. K is the number of latent variables. $\boldsymbol{z} = [1, z^{-1}, \ldots, z^{-q+1}]$, and z^{-1} is the unit delay operator.

Monitoring of the dynamic combination of scores is shown as follows:

$$\boldsymbol{t}_{di}(k) = \boldsymbol{H}_i(z^{-1})\boldsymbol{t}_i(k) = \sum_{j}^{q-1}\boldsymbol{\beta}_{i,j}\boldsymbol{t}_i(k-j), \tag{45}$$

where q is the number of lagged time.

However, given that $\boldsymbol{t}_{di}(k)$ is auto-correlated, performing process monitoring on $\boldsymbol{t}_{di}(k)$ directly may not result in the expected monitoring effect. Accordingly, a VAR model is introduced to describe $\boldsymbol{t}_{di}(k)$.

There are two obvious advantages for VAR models: (1) the model is equipped with multiple explanatory variables, thereby accommodating intricate latent mapping relationships; (2) both short and long-term influences among identical and disparate variables are taken into account, which can potentially enhance the precision of model fitting. Let $\boldsymbol{t}(i) = [\boldsymbol{t}_1(i), \ldots, \boldsymbol{t}_k(i)]^{\mathrm{T}}$, and a new VAR model is established as follows:

$$\boldsymbol{t}(k) = \varphi_1\boldsymbol{t}(k-1) + \cdots + \varphi_p\boldsymbol{t}(k-p) + \boldsymbol{v}(k), \tag{46}$$

where φ_i is the parameter of VAR model, and p is the model order of VAR model. To calculate the model order p and the lagged time q, a cross-validation is utilized.

When modeling $\boldsymbol{v}(k)$ as a zero-mean white noise process, parameters can be efficiently estimated using the multivariate least squares algorithm. Afterward, $\boldsymbol{v}(k)$ is selected as the quality-related dynamic latent variable.

Similarly, the residual component of $\boldsymbol{X}$, which captures variations in quality-unrelated information, is computed as follows:

$$\tilde{\boldsymbol{x}}_s = (\boldsymbol{I} - \boldsymbol{P}_s\boldsymbol{P}_s^{\mathrm{T}})(\boldsymbol{I} - \boldsymbol{P}_d\boldsymbol{P}_d^{\mathrm{T}})(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{R}^{\mathrm{T}})\boldsymbol{X}. \tag{47}$$

Process monitoring indices are constructed as follows:

$$T_y^2 = \boldsymbol{v}(i)^{\mathrm{T}}\Lambda^{-1}\boldsymbol{v}(i), \tag{48}$$

$$Q_s(i) = \left\|\tilde{\boldsymbol{X}}(i)\right\|^2. \tag{49}$$

The control limits L_{T^2} and L_Q corresponding to the statistics conform to the F-squared distribution and the χ^2 distribution, respectively. T_y^2 targets quality-relevant anomalies originating from deviations in normal dynamic patterns. Q_s monitors residual space deviations that may not correlate with quality-related variables.

The complete quality-related dynamic process monitoring strategy is shown in Algorithm 3. To more intuitively evaluate the effectiveness of different methods, the fault detection rate (FDR) is introduced as follows:

$$\mathrm{FDR} = \frac{\text{No. of effective alarms}}{\text{No. of quality} - \text{related faults}} \times 100. \tag{50}$$

For FDR, the effective alarms represent the detected fault related to quality variables. Obviously, high FDR in statistical indicators for quality-related faults is indispensable for the monitoring strategy. The detailed flowchart of the QRDSFA method is shown in Figure 1.

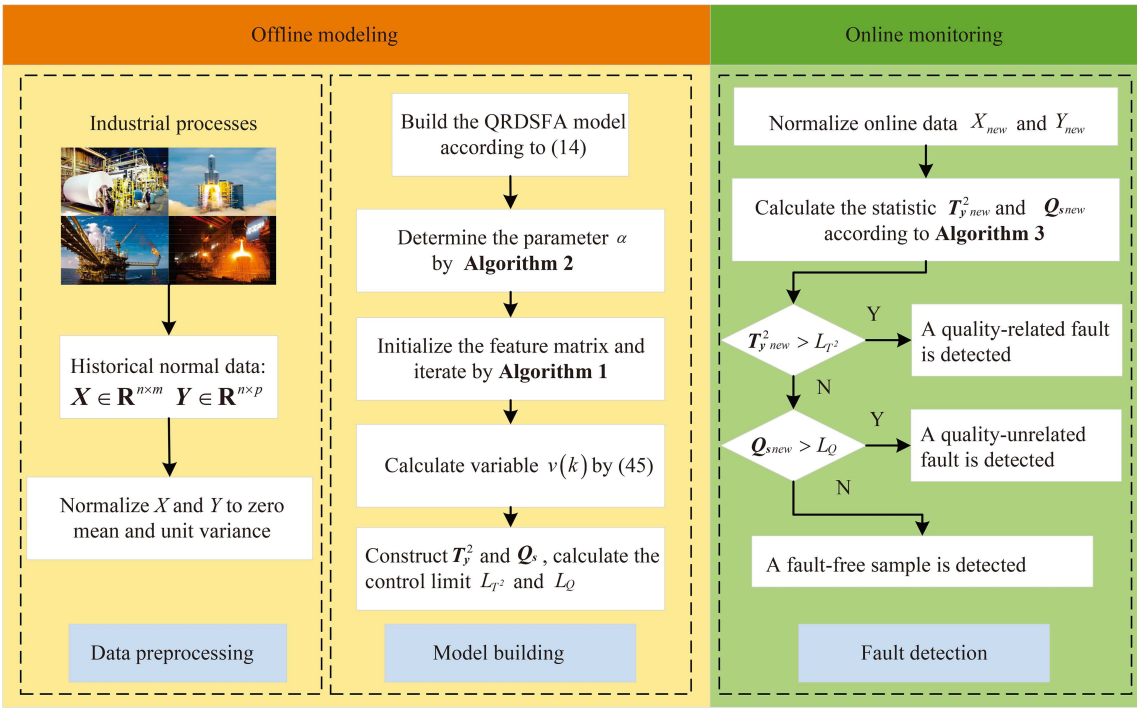

Figure 1 (Color online) Flowchart of the proposed process monitoring method based on QRDSFA.

Algorithm 3 Process monitoring based on QRDSFA.

Offline modeling.

Step 1. Collect input data $\boldsymbol{X}$ and output data $\boldsymbol{Y}$ under normal conditions as training data.

Step 2. Build the QRDSFA model to extract dynamic score $\boldsymbol{t}_i$.

Step 3. Select the parameter α using Algorithm 2.

Step 4. Calculate the quality-related dynamic latent variable $\boldsymbol{v}(k)$ by (45).

Step 5. Construct T_y^2 and Q_s as statistical indicators and calculate the corresponding control limits.

Online monitoring.

Step 1. Collect and normalize test data $\boldsymbol{X}_{new}$ and output data $\boldsymbol{Y}_{new}$.

Step 2. Extract dynamic score $\boldsymbol{t}_i$ by the QRDSFA model.

Step 3. Calculate the quality-related dynamic latent variable $\boldsymbol{v}(k)$ by (45).

Step 4. Construct T_{ynew}^2 and Q_{snew} as statistical indicators.

Step 5. Fault detection logic:

if $T_{ynew}^2 > L_{T^2}$, a quality-related fault is detected;

if $T_{ynew}^2 < L_{T^2}$ and $Q_{snew} \geqslant L_Q$, a quality-unrelated fault is detected;

if $T_{ynew}^2 < L_{T^2}$ and $Q_{snew} < L_Q$, a fault-free sample is detected.

4 Case study

This section evaluates the method's effectiveness using numerical simulations, the Tennessee Eastman process (TEP), and electric servo systems, demonstrating industrial applicability. Meanwhile, several similar monitoring methods including DMPLS [23] and DTPLS [24] are used as comparison algorithms.

It should be noted that although [30] first proposed and improved the idea of combining SFA and PLS, they only applied this idea to the field of data prediction and used RMSE to verify the prediction performance. Although [31] improved this idea, it was still limited to predicting energy efficiency. Considering that the method proposed in this article is mainly used for process monitoring, Refs. [30, 31] are excluded from the comparison algorithms for the time being.

4.1 Numerical simulation

In this subsection, a detailed numerical simulation case study on a dynamic model is conducted, which is described as follows:

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{t}_k + \boldsymbol{e}_k, \\ \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k, \\ \boldsymbol{t}_k = \boldsymbol{a}_1\boldsymbol{t}_{k-1} + \boldsymbol{a}_2\boldsymbol{t}_{k-2} + \boldsymbol{t}_k^*, \end{cases} \tag{51}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0.5586 & 0.2042 & 0.2042 \\ 0.2007 & 0.0492 & 0.4429 \\ 0.0874 & 0.6062 & 0.0664 \\ 0.9332 & 0.5463 & 0.3743 \\ 0.2594 & 0.0958 & 0.2492 \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 0.9249 & 0.4350 \\ 0.6295 & 0.9811 \\ 0.8783 & 0.0960 \\ 0.6417 & 0.5275 \\ 0.7984 & 0.0556 \end{bmatrix}^{\mathrm{T}}, \boldsymbol{t}_k^* = \begin{cases} \boldsymbol{t}_{0_k} + [10;10;10], & 1 \leqslant k < 250, \\ \boldsymbol{t}_{0_k} - [5;5;5], & 250 \leqslant k < 500, \\ \boldsymbol{t}_{0_k} \in N(0, 2^2\boldsymbol{I}_3), & \\ \boldsymbol{t}_{0_k} + [1;1;1]\sin(0.1k), & 500 \leqslant k < 750, \\ \boldsymbol{t}_{0_k}, & 750 \leqslant k < 1000, \end{cases}$$

and

$$\boldsymbol{a}_1 = \begin{bmatrix} 0.4389 & 0.1210 & -0.0862 \\ -0.2966 & -0.0550 & 0.2274 \\ 0.4538 & -0.6573 & 0.4239 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} -0.2998 & -0.1905 & -0.2669 \\ -0.0204 & -0.1585 & -0.2950 \\ 0.1461 & -0.0755 & 0.3749 \end{bmatrix}.$$

Besides, three types of the noise involving $\boldsymbol{e}_k$ and $\boldsymbol{v}_k$ are added as follows:

$$\begin{cases} \text{Noise1:} & \boldsymbol{e}_k^1 \in N(0, 0.05^2\boldsymbol{I}_2),\ \boldsymbol{v}_k^1 \in N(0, 0.05^2\boldsymbol{I}_2) \\ \text{Noise2:} & \boldsymbol{e}_k^2 \in N(0, 0.1^2\boldsymbol{I}_2),\ \boldsymbol{v}_k^2 \in N(0, 0.1^2\boldsymbol{I}_2) \\ \text{Noise3:} & \boldsymbol{e}_k^3 \in 0.1 \times U[(0,1)],\ \boldsymbol{v}_k^3 \in 0.1 \times U[(0,1)], \end{cases} \tag{52}$$

where $\boldsymbol{I}_i$ denotes an i-dimensional vector of unity elements, while $U[(0,1)]$ represents the uniform probability distribution over the interval $[0,1]$.

Under this model framework, 10 replicate experiments are conducted for each noise type. For each experimental run, 1000 normal condition samples are generated, followed by quality-related fault injection according to the protocol below

$$\boldsymbol{t}_k^* = \boldsymbol{t}_k^* + \boldsymbol{\Xi} * f, \tag{53}$$

where $\boldsymbol{\Xi} = [0,0,10]^{\mathrm{T}}$ and $f = 3$.

The optimal parameters of the three methods are as follows: for QRDSFA, the model order $p = 7$, the lagged time $q = 4$, which is calculated by a 2-D cross-validation, as shown in Figure 2. Meanwhile, the number of latent variables $K = 3$. In DMPLS, $A_M = 2$ and $q_M = 1$. In DTPLS, $A_T = 2$ and $q_T = 2$.

The first 200 samples are generated under normal conditions, with the fault introduced starting from the 201st sample. To analyze the characteristics of this augmented dataset, two representative data segments are visualized, where the blue line represents fault-free data y_{free} and the orange line represents faulty data y_{fault}. As shown in Figure 3, compared with y_{free}, the output y_{fault} shows obvious anomalies since the 201st sample. The analysis confirms that the injected fault demonstrates quality-relevant characteristics.

After conducting 20 experiments for each noise case, the average FDRs of the three methods are shown in Table 1. Obviously, QRDSFA has the highest FDR among the three methods, while DTPLS has the lowest FDR. In addition, for Noise1, the results of the 10 repeated experiments are shown in Figure 4. From Figure 4, QRDSFA shows both excellent and stable performance, with the FDR close to 100%. In the meantime, DMPLS and DTPLS have a relatively poor performance.

To provide a more intuitive demonstration of the three methods' effectiveness, the results of one repeated experiment are shown in Figures 5–7. The specific FDRs of QRDSFA, DMPLS, and DTPLS are 95.81%, 72.38%, and 32.79%, respectively. Furthermore, as evident from the provided T_y^2 and Q_s statistics, QRDSFA demonstrates the capability to detect nearly all quality-related faults subsequent to the 201st sample. In contrast, both DMPLS and DTPLS exhibit a significant number of missed alarms below the threshold.

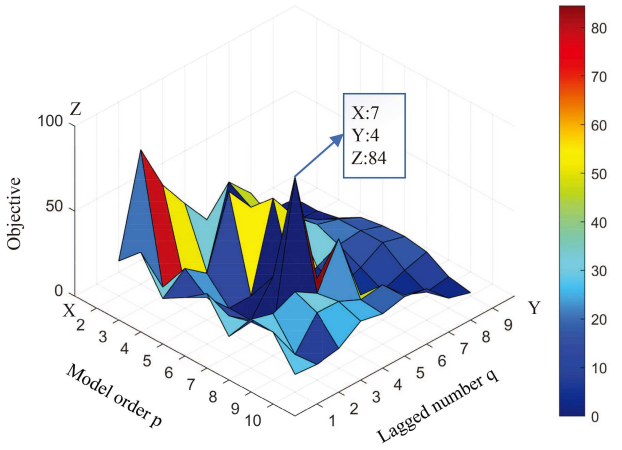

Figure 2 (Color online) 2-D cross-validation results for numerical simulation.

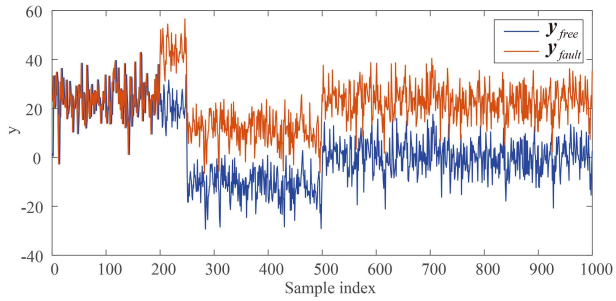

Figure 3 (Color online) The variation of y under quality-related fault and fault-free state.

Table 1 The average FDRs and standard deviation of three noise cases (%).

Method	QRDSFA	DMPLS	DTPLS
Noise1	91.24	64.70	33.37
Noise2	94.08	66.75	31.52
Noise3	93.81	63.44	37.87

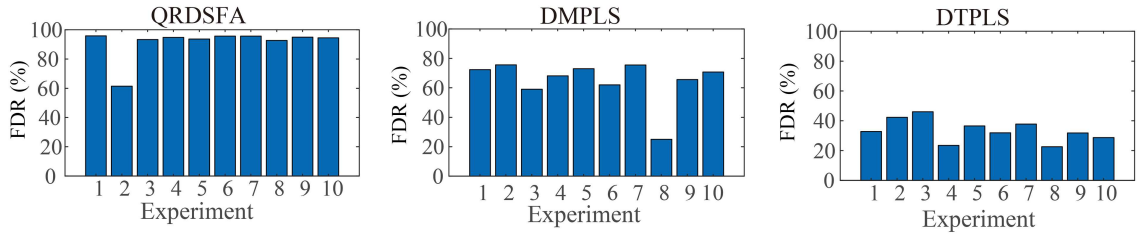

Figure 4 (Color online) FDR plot for three methods on Noise1 (10 experimental runs).

4.2 Tennessee Eastman process

The TEP represents a widely recognized benchmark derived from an actual chemical industrial process. Characterized by time-varying parameters, strong inter-variable couplings, and hybrid dynamic behaviors, this process has become a standard dataset for research in process control, monitoring, and fault detection [20, 22].

The process features two distinct variable blocks: the XMV block containing 11 manipulated variables and the XMEAS block comprising 41 measured variables. In the experiment, variables XMEAS (1–22) and XMV (1–11) are chosen as $\boldsymbol{X}$, while XMEAS 35 is chosen as $\boldsymbol{Y}$. In addition, there are 21 fault cases within the complete process, including 15 known faults denoted by IDV(1–15). For validation in the TEP context, it is essential to determine whether the fault correlates with $\boldsymbol{Y}$. As established in [34], IDV indices (1, 2, 5–8, 12–14) are categorized as quality-related faults, while indices (3, 4, 9, 11, 15) are classified as quality-unrelated faults. In total, the training dataset comprised 960 normal condition samples for offline modeling. Subsequently, different faulty data sets with all the faults occurring at the 161st sample are introduced for online monitoring. In QRDSFA, $K = 6$, and the time delay is $q = 1$. In DM-PLS, $A_M = 2$ and $q_M = 3$. In DTPLS, $A_T = 4$ and $q_T = 3$. These parameters were optimized through a two-dimensional cross-validation methodology to ensure robust parameter estimation.

For quality-related faults in the TEP, the FDRs of QRDSFA, DMPLS and DTPLS are shown in Table 2, where the highest FDR for each fault type is bolded for emphasis. From Table 2, it can be seen that QRDSFA has the highest FDRs for IDV(1, 2, 7, 8, 12, 13). Meanwhile, QRDSFA demonstrates comparable fault detection capabilities to DMPLS for IDV(6) and to DTPLS for IDV(14). DMPLS shows suboptimal performance on IDV(7) and IDV(14), with significant missing alarm rates: 18.73% FDR for IDV(7) and 23.97% FDR for IDV(14). IDV(8) is a typical quality-related fault, characterized by a random variation fault on the feed composition.

For IDV(8), the monitoring results by the three methods are shown in Figure 8, with the solid blue line rep-

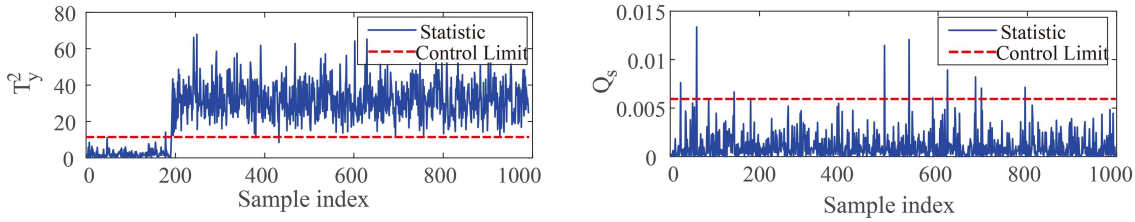

Figure 5 (Color online) Quality-related fault detection results obtained by QRDSFA.

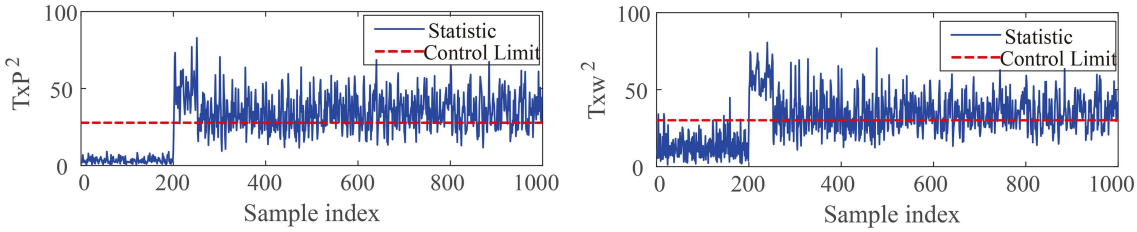

Figure 6 (Color online) Quality-related fault detection results obtained by DMPLS.

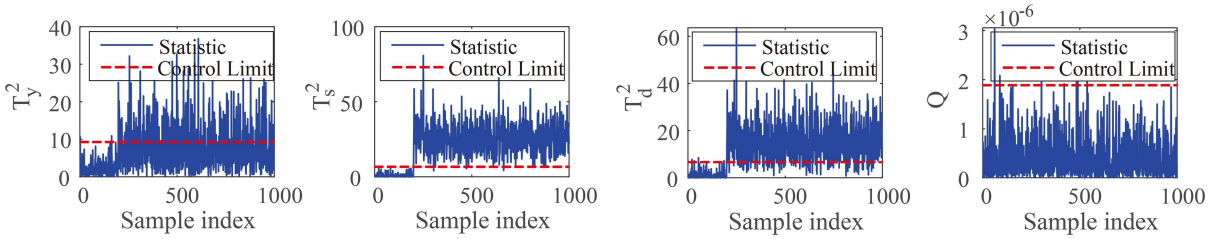

Figure 7 (Color online) Quality-related fault detection results obtained by DTPLS.

Table 2 FDRs of quality-related faults in the TEP (%).

Fault ID	Fault description	QRDSFA	DMPLS	DTPLS
IDV(1)	Feed ratio	**99.50**	96.13	91.64
IDV(2)	Ration constant	**98.50**	87.27	95.63
IDV(6)	Feed loss	99.37	**99.63**	97.75
IDV(7)	C header pressure loss	**86.73**	18.73	65.54
IDV(8)	Feed composition	**87.98**	61.42	66.91
IDV(12)	Cooling water temperature	**89.36**	81.65	82.27
IDV(13)	Reaction kinetics	**93.87**	83.52	89.14
IDV(14)	Reactor cooling water	99.55	23.97	**99.63**

resenting the monitoring statistic and the dashed red line indicating the control limit. The figure demonstrates that DMPLS and DTPLS show significant missing alarms starting from the 161st sample point. In comparison, QRDSFA demonstrates superior performance in quality-related fault detection.

4.3 Electric servo systems

The electric servo systems are mainly composed of servo controllers, drivers, electromechanical actuators, and power supplies [35]. By precisely controlling parameters such as motor speed, position, and acceleration, precise control of mechanical motion is achieved, as shown in Figure 9.

By analyzing the working principle of electric servo mechanisms, typical fault types can generally be classified into three categories: environmental factor faults, mechanical system faults, and electrical system faults. (1) Environmental factor faults. For control circuits, DSP processing units are both core components and typical weak links. Under long-term operation, thermal stress cycles caused by temperature changes may lead to cracks or damage inside the DSP, manifested as output voltage jumps. (2) Mechanical system faults. The electromechanical actuator, in response to the position command issued by the servo driver, drives the servo motor to rotate bidirectionally, thereby causing the ball screw to move back and forth in a straight line. Under long-term op-

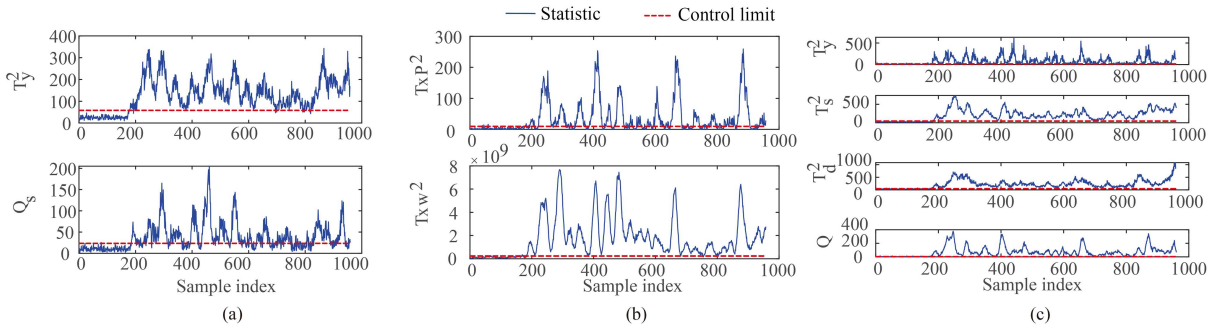

Figure 8 (Color online) Detection results of IDV(8) obtained by QRDSFA (a), DMPLS (b), and DTPLS (c).

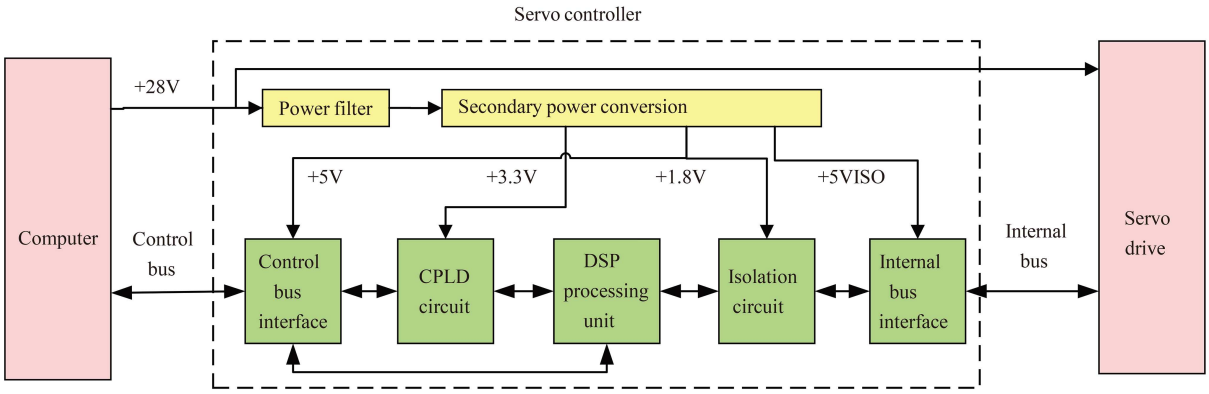

Figure 9 (Color online) A schematic layout of the electric servo systems.

eration, mechanical wear may lead to abnormal output of electromechanical actuators. (3) Electrical system faults. Permanent magnet synchronous motors in servo systems rotate precisely by processing control signals and dynamically adjusting based on real-time feedback. Under long-term operation, magnet failure may lead to a decrease in the speed of the servo motor.

In the practical operation of the electric servo mechanisms, the actuator plays an important role and exhibits a high susceptibility to the aforementioned three types of faults. After prolonged use, mechanical wear and slow evaporation of grease from the actuator can lead to a decrease in work efficiency, which in turn causes changes in position feedback parameters, ultimately resulting in changes in actuator displacement. Based on the electric servo mechanisms experimental platform, a set of normal data and quality-related fault data of the actuator are collected, with 10 dimensions of process variables and 1 dimension of quality variables shown in Table 3. The total number of samples in the dataset is 500.

In the subsection, the process variables are set as $\boldsymbol{X}$ and the quality variables are set as $\boldsymbol{Y}$. Considering that the input is a sine signal, the quality variable also follows the input signal to exhibit sine periodic changes, as shown in Figure 10. The visualization in the figure shows the quality variable from the normal dataset as a solid blue line and from the faulty dataset as a solid green line. In addition, the red dashed line demarcates the threshold, with samples exceeding this boundary indicating the range of faulty samples.

For the electric servo mechanisms, the process monitoring effectiveness of QRDSFA, DMPLS and DTPLS is shown in Table 4, where the highest FDR and the lowest FAR are boldfaced. The monitoring results are shown in Figure 11, with the solid blue line representing the monitoring statistic and the dashed red line indicating the control limit. In QRDSFA, $K = 9$, and the time delay is $q = 1$. In DMPLS, $A_M = 6$ and $q_M = 3$. In DTPLS, $A_T = 4$ and $q_T = 1$.

5 Some discussions

5.1 Ablation experiments

In this subsection, we perform ablation experiments to systematically evaluate the contribution of the VAR model by removing its components from the proposed framework. To ensure the persuasiveness of the experimental

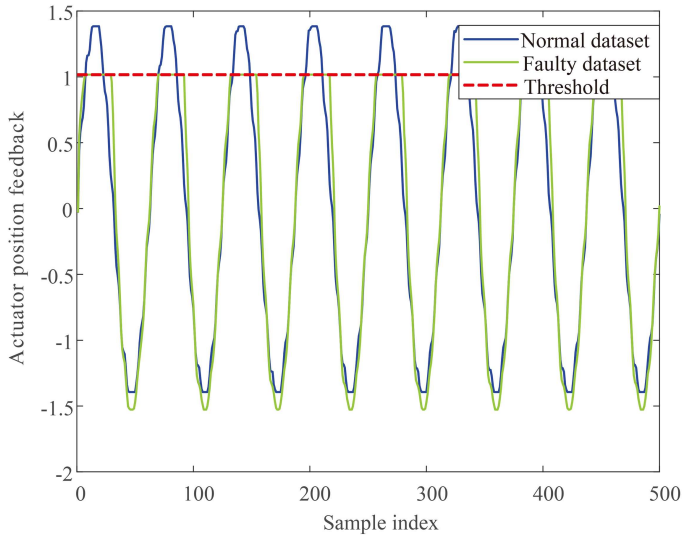

Figure 10 (Color online) Actuator position feedback.

Table 3 Monitoring variables of the electric servo systems.

No.	Variable	Type	Description
1	δf	Process variables	Motor equivalent position feedback
2	N	Process variables	Speed feedback
3	Iq	Process variables	Q-axis current feedback
4	Id	Process variables	D-axis current feedback
5	δc	Process variables	Nozzle swing angle command
6	Rdc	Process variables	Motor angle
7	$V(+5)$	Process variables	+5 V power supply status
8	$V(-5)$	Process variables	−5 V power supply status
9	$V(+10)$	Process variables	+10 V power supply status
10	$V(-10)$	Process variables	−10 V power supply status
11	δt	Quality variables	Actuator position feedback

Table 4 Performance of different methods of the electric servo systems (%).

Indices	QRDSFA	DMPLS	DTPLS
FDR	99.15	94.11	98.31

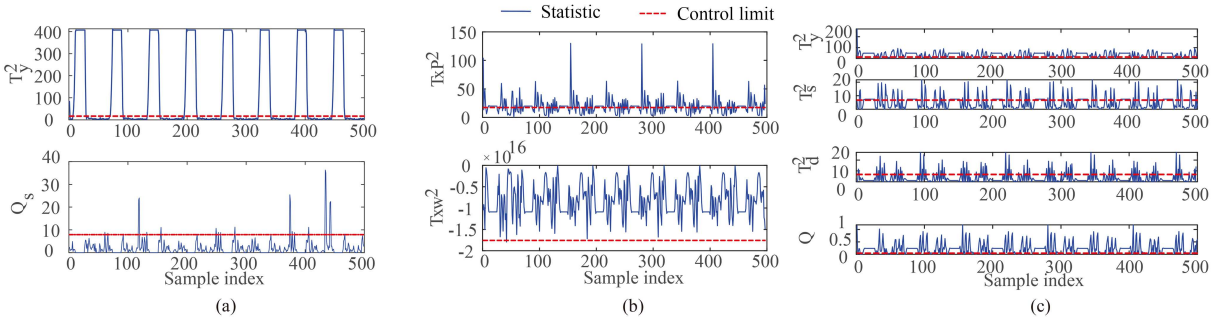

Figure 11 (Color online) Detection results of the electric servo mechanisms obtained by QRDSFA (a), DMPLS (b), and DTPLS (c).

findings, we introduce two key measures. First, three additional numerical simulation cases (Noise 4–6) are selected as testing benchmarks, extending the scenarios presented in Subsection 4.1. Second, two comparative approaches are adopted: the original QRDSFA method and an ablated version wherein the VAR components are replaced with AR (autoregressive) counterparts.

The ablation study results are summarized in Table 5, with the FDRs from 10 repeated experiments under

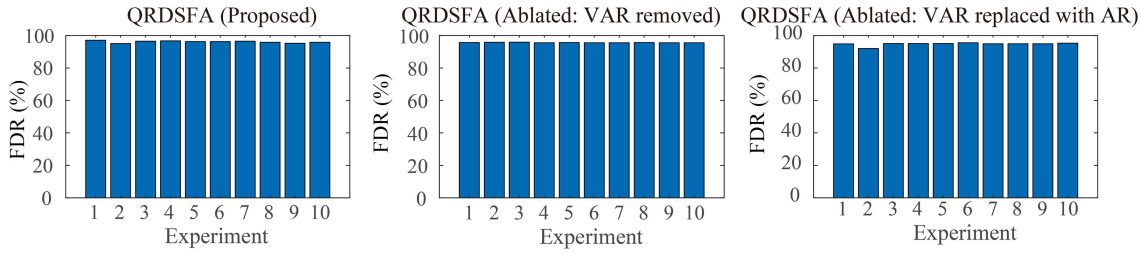

Figure 12 (Color online) FDR plot for three methods on Noise4 (10 experimental runs).

Table 5 The average FDRs of three noise cases (%).

Method	QRDSFA (proposed)	QRDSFA (ablated: VAR removed)	QRDSFA (ablated: VAR replaced with AR)
Noise4	96.15	95.67	94.80
Noise5	96.47	95.73	95.31
Noise6	96.55	95.79	95.36

Noise4 conditions further illustrated in Figure 12. Although the performance gaps among QRDSFA (proposed), QRDSFA (ablated: VAR removed), and QRDSFA (ablated: VAR replaced with AR) appear marginal, the proposed method consistently achieves the highest FDR across all three numerical simulation cases. This subtle yet consistent superiority highlights the advantage conferred by the VAR module. The results confirm that the incorporation of the VAR component effectively capturing cross-variable temporal dependencies leads to measurable performance improvements, underscoring its essential role in enhancing fault detection capability within the QRDSFA framework.

5.2 Algorithm complexity analysis

To evaluate the industrial applicability of the proposed QRDSFA algorithm, this subsection analyzes its time and space complexity. The complexity is primarily concentrated in the model training phase, while the online monitoring phase exhibits extremely low complexity, making it suitable for real-time applications.

5.2.1 *Time complexity analysis*

The computational burden of QRDSFA is primarily concentrated in the offline training phase, which is acceptable for an industrial application where a model is built once and then deployed. The complexity of major steps is analyzed as follows.

(1) Objective function. The core computational burden comes from solving the combined optimization problem (13), especially the construction of $\boldsymbol{X}_d^{\mathrm{T}}\boldsymbol{Y}$ and matrix multiplication. Considering that $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, output matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$ and iteration count I_i, the overall time complexity is roughly $O(npI_im)$.

(2) Integrated decomposition. For DPLS modeling, the main time complexity is $O(mn)$; for SFA, the main time complexity is $O(nm^2 + m^3)$ (covariance matrix and SVD operation). Therefore, the overall time complexity is roughly $O(nm^2 + m^3)$.

(3) VAR modeling. The vector autoregression modeling maintains $O(mq^2)$ complexity for residual dynamics characterization, where q is the lagged time.

The overall computational complexity of the QRDSFA offline training phase can be synthesized from the complexities of its major components as follows: the total time complexity is dominated by the most computationally expensive operations, which is shown as $O(npI_im + nm^2 + m^3 + mq^2)$. Considering that m^3 is the dominant term for systems with a large number of variables, and npI_im dominates for large datasets, the overall asymptotic time complexity can be expressed as $O(npI_im + m^3)$.

5.2.2 *Space complexity analysis*

The memory requirements are influenced by the integrated nature of the approach.

(1) Covariance storage. Requires $O(n^2)$ space for storing the combined covariance matrices that incorporate both DPLS and SFA criteria through α.

(2) Optimization memory. The self-search strategy and Lagrange multiplier implementation need $O(mn + Kn)$ additional memory for storing intermediate optimization variables.

The total space complexity is $O(n^2 + mn + Kn)$.

Table 6 Algorithm complexity analysis of QRDSFA.

Complexity type	Phase	Phase complexity	Cumulative complexity
Time complexity	Objective function	$O(npI_im)$	$O(npI_im + m^3)$
	Integrated decomposition	$O(nm^2 + m^3)$	
	VAR modeling	$O(mq^2)$	
Space complexity	Covariance storage	$O(n^2)$	$O(n^2 + mn + Kn)$
	Optimization memory	$O(mn + Kn)$	

5.2.3 *Overall complexity analysis*

As shown in Table 6, the computational analysis demonstrates that the proposed QRDSFA algorithm achieves superior monitoring performance while maintaining computational feasibility. Although the joint optimization introduces an additional complexity term $O(npI_im)$ compared to conventional DMPLS ($O(nm^2 + m^3)$) and DTPLS ($O(nlm^2)$) methods, this overhead is acceptable for offline training and enables precise quality-related dynamic feature extraction. Notably, the online monitoring complexity remains at $O(m^2)$, ensuring real-time implementability in industrial environments.

5.3 Quality-related dynamic information retention ratio

To quantitatively evaluate the ability of the QRDSFA algorithm to retain quality-related information in the extracted features, this subsection introduces a new metric: quality-related dynamic information retention ratio (QDIRR). The core foundation of this metric lies in the mutual information (MI) between the feature variables and the quality variable.

5.3.1 *Definition of mutual information*

Mutual information originates from information theory and is used to measure the degree of interdependence between two random variables. Specifically, the mutual information between a feature variable T and the quality variable Y is defined as

$$\mathrm{MI}\left(T;Y\right)=\sum_{y\in Y}\sum_{t\in T}p\left(t,y\right)\log\frac{p\left(t,y\right)}{p\left(t\right)p\left(y\right)},\tag{54}$$

where $p\left(t\right)$ and $p\left(y\right)$ are the marginal probability distributions of T and Y, respectively; and $p\left(t,y\right)$ is their joint probability distribution. The logarithm is generally base 2, and the unit is bits.

5.3.2 *Calculation of QDIRR*

Based on mutual information, the QDIRR metric is defined as the ratio of the quality-related information retained by the first K selected features to the total quality-related information. Its calculation formula is as follows:

$$\mathrm{QDIRR}\left(K\right)=\frac{\sum_{i=1}^{K}\mathrm{MI}\left(T_i;Y\right)}{\sum_{i=1}^{m}\mathrm{MI}\left(T_i;Y\right)},\tag{55}$$

where $\mathrm{MI}\left(T_i;Y\right)$ represents the mutual information between the i-th feature variable and the quality variable Y, K is the number of features selected by the algorithm, and m is the total number of features. QDIRR is not only a computational metric but also possesses a solid mathematical foundation and clear physical significance. Its key characteristics are analyzed below.

(1) Range and boundary conditions. According to the definition, when the first K selected features are completely independent of the quality variable Y, i.e., $\mathrm{MI}\left(T_i;Y\right)=0,\forall i\in[1,K]$, then $\mathrm{QDIRR}\left(K\right)=0$. This indicates that the selected features contain no quality-related information. When $K=m$, i.e., all features are selected, $\mathrm{QDIRR}\left(m\right)=1$, indicating that all quality-related information in the data has been retained. Thus, the value range of QDIRR is $[0,1]$.

(2) Curve trend. A typical QDIRR curve exhibits a monotonically increasing trend. In the initial stage, the curve has a steep slope, indicating that adding the first few features yields significant information gain. As K increases, the curve gradually flattens, and the slope decreases, reflecting diminishing marginal gains from additional features. This occurs because the feature extraction algorithm prioritizes the most informative features.

(3) Theoretical link to monitoring performance. A higher QDIRR value indicates that the core quality-related dynamic information is more completely preserved in the features used by the monitoring model. Theoretically,

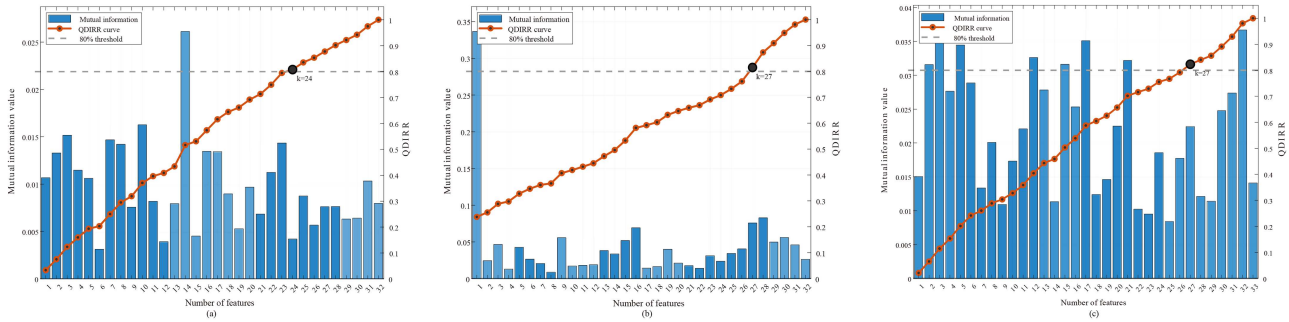

Figure 13 (Color online) QDIRR of IDV(2) obtained by QRDSFA (a), DMPLS (b), and DTPLS (c).

Table 7 Performance of different methods of IDV(2) (%) at the same QDIRR.

Indices	QRDSFA	DMPLS	DTPLS
Number of features	24	27	27
FDR	98.50	85.35	92.56

when different methods retain sufficient dynamic information at the same QDIRR level, the method that achieves superior monitoring performance with fewer features is considered more desirable.

5.3.3 *Experimental validation*

To validate the effectiveness of the QDIRR metric, experiments were conducted using the IDV(2) fault data from the Tennessee Eastman process described in Subsection 4.2. The validation procedure is as follows: first, the QDIRR values corresponding to different numbers of features K were calculated for each algorithm; second, a unified QDIRR threshold (80%) was set to determine the special number of features K required by each algorithm; finally, the fault detection performance of each algorithm was evaluated and compared under the condition of using K features.

The QDIRR curves of the three methods for the IDV(2) fault data are shown in Figure 13. It can be observed that when the preset QDIRR threshold is 80%, the required number of features K for QRDSFA, DMPLS, and DTPLS are 24, 27, and 27, respectively. Under this configuration, the fault detection results of the algorithms are presented in Table 7. The results demonstrate that, at the same QDIRR level, the proposed QRDSFA method requires the fewest features while achieving the best fault detection performance.

5.4 Algorithm robustness analysis

To validate the applicability and effectiveness of the proposed QRDSFA method in practical industrial scenarios, this section systematically evaluates its robustness under non-Gaussian noise conditions. The experiments are based on the numerical simulation case described in Subsection 4.1. A mixture Gaussian noise model is introduced to simulate non-ideal noise conditions commonly found in industrial data. The performance stability of the QRDSFA method is quantitatively analyzed based on the evolution trend of FDR.

To accurately simulate typical impulsive disturbances in industrial processes, a mixture Gaussian noise is introduced to the observations of both process variables and quality variables in the simulation. Specifically, the original equation (52) is modified as follows:

$$\text{Noise } R: \begin{cases} \boldsymbol{e}_k^R \sim \left[(1-\theta)\times N\left(0,\delta_1^2\right)+\theta\times N\left(0,\delta_2^2\right)\right], \\ \boldsymbol{v}_k^R \sim \left[(1-\theta)\times N\left(0,\delta_1^2\right)+\theta\times N\left(0,\delta_2^2\right)\right], \end{cases} \tag{56}$$

where $N\left(0,\sigma^2\right)$ represents a Gaussian distribution with zero mean and variance δ; δ_1 and δ_2 denote the standard deviations of the background noise and the impulsive noise, respectively; $\delta_2 = 10\delta_1$; the parameter θ represents the contamination ratio of the impulsive noise. For a comprehensive evaluation, the contamination ratio θ is varied from 0.1 to 0.9 in increments of 0.1, thereby covering a range of practical industrial scenarios from mild to severe contamination. The corresponding FDR performance of the QRDSFA method under different contamination ratios is summarized in Figure 14.

Analysis of the results presented in Figure 14 leads to two key conclusions. (1) Remarkable absolute performance. Even under extreme noise contamination ($\theta = 0.9$), the FDR of QRDSFA remains at a high level of 86.85%. This robust performance convincingly demonstrates the method's reliable fault detection capability in the presence of

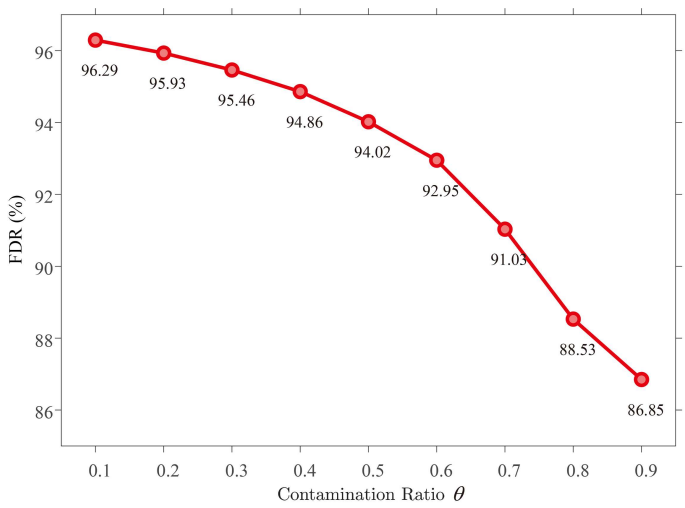

Figure 14 (Color online) FDR plot corresponding to QRDSFA method under different contamination ratios θ.

severe non-Gaussian noise. (2) Graceful performance degradation. As the contamination ratio θ increases from 0.1 to 0.9, the FDR smoothly decreases from 96.29% to 86.85%, exhibiting a gradual and well-controlled declining trend. This characteristic of "graceful degradation" is a crucial indicator of strong robustness, signifying that the performance of QRDSFA does not collapse abruptly with increasing noise intensity but maintains predictable stability.

Further analysis reveals that this pronounced robustness stems from two core design principles of the QRDSFA framework. On the one hand, the slow feature extraction mechanism focuses on the most gradually varying dynamic features within the process variables. This allows it to naturally identify and categorize the fast, impulsive disturbances characteristic of non-Gaussian noise as insignificant "fast-varying" information, thereby effectively filtering or suppressing them into the residual space. On the other hand, the quality-related projection framework ensures that the feature subspace decoupled from the complex dynamics maintains strong relevance to the final product quality variables. This design makes the monitoring statistics derived from this subspace remarkably insensitive to the interference of non-Gaussian noise.

6 Conclusion

In this article, a novel QRDSFA method is proposed to achieve effective process monitoring. On the basis of DPLS and SFA, a new objective function is established to extract comprehensive quality-related dynamic characteristics. Afterward, by using the Lagrange multipliers method, the objective function is solved, and the model parameters are determined. Furthermore, a VAR model is introduced to detect anomalies in the dynamic process at the current moment. Finally, statistical indicators are constructed based on latent variables and residuals to monitor dynamic changes related to quality.

According to the simulation and application studies in Sections 4 and 5, the proposed QRDSFA method has a better dynamic process monitoring performance than other compared methods. However, the adaptability and robustness of the proposed method have not been discussed in the article, which will be studied in the future.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62273354, 62227814).

References

1 Zhou L, Wang Y X, Ge Z Q. Multi-rate principal component regression model for soft sensor application in industrial processes. Sci China Inf Sci, 2020, 63: 149205
2 Huang K K, Tao S J, Liu Y S, et al. Label propagation dictionary learning based process monitoring method for industrial process with between-mode similarity. Sci China Inf Sci, 2022, 65: 110203
3 Obanya P O, Coetzer R L J, Olivier C P, et al. Variable contribution analysis in multivariate process monitoring using permutation entropy. Comput Ind Eng, 2024, 190: 110064
4 Wu D, Zhou D, Chen M. Performance-driven component selection in the framework of PCA for process monitoring: a dynamic selection approach. IEEE Trans Contr Syst Technol, 2022, 30: 1171–1185
5 Li H, Jia M, Mao Z. Dynamic reconstruction principal component analysis for process monitoring and fault detection in the cold rolling industry. J Process Control, 2023, 128: 103010
6 Cheng C, Qiao X, Teng W, et al. Principal component analysis and belief-rule-base aided health monitoring method for running gears of high-speed train. Sci China Inf Sci, 2020, 63: 199202
7 Luo Z, Li H. Dynamic partial-least-squares-based fault detection for nonlinear distributed parameter systems. IEEE Trans Instrum Meas, 2024, 73: 3515509

8 Yang J, Wang J, Ye Q, et al. A novel fault detection framework integrated with variable importance analysis for quality-related nonlinear process monitoring. Control Eng Pract, 2023, 141: 105733
9 Hasnen M, Zabiri H, Ammar A. Semi-supervised adaptive PLS soft-sensor with PCA-based drift correction method for online valuation of NOx emission in industrial water-tube boiler. Process Saf Environ Prot, 2023, 172: 787–801
10 Kong X, Liu M, Zhang Q, et al. Semisupervised kernel independent component analysis and its application for fault diagnosis. IEEE Trans Instrum Meas, 2025, 74: 3508914
11 Yang N C, Ismail H. Robust intelligent learning algorithm using random forest and modified-independent component analysis for PV fault detection: in case of imbalanced data. IEEE Access, 2022, 10: 41119–41130
12 Liu M, Kong X, Luo J, et al. Incipient fault detection based on ensemble learning and distribution dissimilarity analysis in multi-feature processes. Meas Sci Technol, 2024, 35: 045905
13 Chen Q, Wang Y. Key-performance-indicator-related state monitoring based on kernel canonical correlation analysis. Control Eng Pract, 2021, 107: 104692
14 Qin S J, Zheng Y. Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. AIChE J, 2012, 59: 496–504
15 Zheng J, Zhao C. Enhanced canonical variate analysis with slow feature for dynamic process status analytics. J Process Control, 2020, 95: 10–31
16 Han H, Zhao D, Gao X. A data-driven key performance indicator-related monitoring scheme for dynamic nonlinear systems. IEEE Trans Circuits Syst II Exp Briefs, 2024, 71: 2074–2078
17 Chen Y, Wang D. An improved deep kernel partial least squares and its application to industrial data modeling. IEEE Trans Ind Inf, 2024, 20: 7894–7903
18 Yu W, Lee S J, Cho H. Partial least squares regression trees for multivariate response data with multicollinear predictors. IEEE Access, 2024, 12: 36636–36644
19 Chu F, Mo S, Hao L, et al. Operating performance assessment of complex nonlinear industrial process based on kernel locally linear embedding PLS. IEEE Trans Automat Sci Eng, 2024, 21: 593–605
20 Hu C, Luo J, Kong X, et al. Orthogonal multi-block dynamic PLS for quality-related process monitoring. IEEE Trans Automat Sci Eng, 2024, 21: 3421–3434
21 Ku W, Storer R H, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis. Chemom Intell Lab Syst, 1995, 30: 179–196
22 Shang C, Huang X, Suykens J, et al. Enhancing dynamic soft sensors based on DPLS: a temporal smoothness regularization approach. J Process Control, 2015, 28: 17–26
23 Jiao J, Yu H, Wang G. A quality-related fault detection approach based on dynamic least squares for process monitoring. IEEE Trans Electron Devices, 2016, 63: 2625–2628
24 Li G, Liu B, Qin S, et al. Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: the dynamic T-PLS approach. IEEE Trans Neural Netw, 2011, 22: 2262–2271
25 Russell E L, Chiang L H, Braatz R D. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemom Intell Lab Syst, 2000, 51: 81–93
26 Wiskott L, Sejnowski T J. Slow feature analysis: unsupervised learning of invariances. Neural Comput, 2002, 14: 715–770
27 He Y, Guan Z, Wang J. Virtual sensing techniques for nonstationary processes based on a multirate probabilistic dual-latent-variable supervised slow feature analysis. IEEE Trans Ind Inf, 2024, 20: 4884–4893
28 Zhang J, Zhou D, Chen M, et al. Continual learning-based probabilistic slow feature analysis for monitoring multimode nonstationary processes. IEEE Trans Automat Sci Eng, 2024, 21: 733–745
29 Shang C, Yang F, Huang B, et al. Recursive slow feature analysis for adaptive monitoring of industrial processes. IEEE Trans Ind Electron, 2018, 65: 8895–8905
30 Chiplunkar R, Huang B. Output relevant slow feature extraction using partial least squares. Chemom Intell Lab Syst, 2019, 191: 148–157
31 Zhu L, Li Z, Chen J. Evaluating and predicting energy efficiency using slow feature partial least squares method for large-scale chemical plants. Energy, 2021, 230: 120582
32 Qiao K, Yu K, Qu B, et al. An evolutionary multitasking optimization framework for constrained multiobjective optimization problems. IEEE Trans Evol Comput, 2022, 26: 263–277
33 Shang C, Yang F, Gao X, et al. Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis. AIChE J, 2015, 61: 3666–3682
34 Zhang Y, Wang W, Deng Y, et al. Signal reconstruction algorithm for azimuth multichannel SAR system based on a multiobjective optimization model. IEEE Trans Geosci Remote Sens, 2020, 58: 3881–3893
35 Zhou D, Li G, Qin S J. Total projection to latent structures for process monitoring. AIChE J, 2010, 56: 168–178