Recursive-CPLS-based quality-relevant and process-relevant fault monitoring with application to the Tennessee Eastman process

CHANGHUA HU1, ZHONGYING XU1, XIANGYU KONG1, AND JIAYU LUO1
1Xi’an Research Institute of High Technology, Xi’an 710025, China

This work was supported in part by the National Natural Science Foundation of China under Grants 61673387, 61374120, 61903375 and 61833016.

ABSTRACT In industrial processes, the quality of the product is crucial. The batch partial least squares (PLS) monitoring model can effectively monitor for quality-related faults. In process monitoring, to overcome time-varying disturbances, the monitoring model needs to be updated regularly. Efficiently updating the monitoring model represents a serious problem. This paper proposes a recursive concurrent projection to latent structures (RCPLS) algorithm, which can both update models more efficiently with historical model parameters and new data and provide better quality-related fault monitoring results than can static concurrent projection to latent structures (CPLS). Based on RCPLS, a complete set of process monitoring technologies is proposed. These technologies can automatically filter and store modelable data and adaptively update the online monitoring model. The updated computational quantities of the RCPLS model and the CPLS model are compared through the Tennessee Eastman process (TEP). The effectiveness of the RCPLS algorithm is verified, and a comprehensive comparison of the quality-related fault detection capabilities of RCPLS and CPLS is performed. The results show that RCPLS can significantly reduce the computational burden and increase the monitoring performance.

INDEX TERMS Projection to latent structure, Process monitoring, Quality-related, Model updating.

I. INTRODUCTION

In industrial process monitoring, online fault detection is a challenging task. With the accumulation of large amounts of industrial data, multivariate statistical process monitoring methods, which are based on principal component analysis (PCA) [1], partial least squares (PLS) [2], Fisher discriminant analysis (FDA) [3], and independent component analysis (ICA) [4], have become widely used. Qin [5] reviews the methods and applications of data-driven fault detection and diagnosis developed over the last two decades. Ge [6,7] provides a systematic review of data-driven modeling and monitoring for plant-wide processes. Given the high-dimensional nature of data in the process industry, Ge [8] conducts a tutorial review of probabilistic latent variable models for process data analytics. Focusing on industrial process data, which usually suffer from missing data and include outliers, Zhu [9] provides a systematic review of various state-of-the-art data preprocessing tricks. From the perspective of machine learning, Ge [10] provides a review of existing data mining and analytics applications.

In industrial production, the quality of products is often the focus of attention [11]. Therefore, quality-related information needs to be extracted from the process data and monitored [12]. The PLS, also called projection to latent structure, model can precisely extract the information related to the quality variable $Y$ in the process variable $X$. Therefore, PLS is highly advantageous for quality-related fault detection [13]. However, the traditional PLS model does not decompose $X$ orthogonally, and PLS suffers from inappropriate monitoring indicators of the $X$ residuals in process monitoring [14]. Zhou [15] proposes the total projection of the latent structure (TPLS) model, which effectively overcomes the shortcomings of traditional PLS. Modified PLS (MPLS), proposed by Yin and Ding [16], can also effectively overcome the shortcomings of traditional PLS. However, TPLS still fails to monitor parts with unpredictable quality because it divides
To extend the scope of application of PLS, scholars have introduced the advantages of different algorithms into PLS. Data normally processed by the PLS algorithm are data under Gaussian conditions. To address the non-Gaussian properties of dynamic variables, Zhang [11] combines the advantages of dynamic PLS and ICA and proposes the ICA-DPLS algorithm. To effectively extract the local structure information of the data, Zhong [23] proposes a quality-related global and local partial least squares (QGLPLS) model by combining the advantages of LPP and PLS. To better focus on the locality-preserving characteristics, a new integration method called the locality-preserving partial least squares (LPPLS) model is proposed by Wang [24]. Recently, Zhou [25] proposed the global-plus-local projection to latent structures (GPLPLS) model. GPLPLS, which ensures that the correlations among the data after dimensional reduction are still the highest, can accurately distinguish between quality-recoverable faults, quality-unrelated faults and minor quality-related faults.

However, these algorithms are all batch processing algorithms, and industrial processes are often characterized by time-varying changes such as the degradation of catalytic performance, drift of operating points, and efficiency degradation. Therefore, it is necessary to periodically update the monitoring model with newly measured process data so that the model can reflect these changes. In the traditional batch model updating method, the number of samples of the modeling data continuously increases; this greatly increases the number of model update calculations. To update the model more quickly, Helland [26] proposes a recursive partial least squares (RPLS) algorithm.

The RPLS algorithm is different from the traditional batch processing algorithms described above. It can use the old model parameters and new data to update the model instead of repeatedly using old data with a large number of samples. This method can greatly reduce the computational burden of model updating and improve the efficiency of model updating. Later, Wold [27] introduced the concept of exponentially weighted moving average (EWMA) to PCA and PLS models and proposed a simpler model update method. Dayal and MacGregor [28] propose a recursive exponentially weighted PLS algorithm that does not use all historical data for adaptive control and prediction. To further enhance the effectiveness of RPLS, Qin [29] propose a block recursive PLS (BRPLS) and updated the model with new and old model parameters. Moreover, Qin introduced a sliding window and adaptive forgetting factor technology into BRPLS so that the updated model can better adapt to changes in the process. RPLS has been widely used. Recently, Dong [30] introduced the idea of RPLS into TPLS, proposed a recursive TPLS (RTPLS), and demonstrated the validity of the new model through TEP.

Similar to TPLS, RTPLS cannot monitor parts with unpredictable quality. To effectively update the model and more comprehensively monitor quality-related faults, this paper proposes a recursive concurrent projection to latent structures (RCPLS) model. RCPLS fully utilizes the advantages of the CPLS model for the comprehensive monitoring of the process. It also provides the advantages of the high efficiency of the RPLS model updating and enables a more comprehensive quality-related fault detection function. Based on the Tennessee Eastman process (TEP), a comparison of RCPLS and CPLS calculations is performed, herein highlighting the advantages of RCPLS in terms of model updating. The effectiveness of the RCPLS algorithm is verified through quality-related and process-related faults. The quality-related fault capabilities of RCPLS and CPLS are compared.

The remainder of this paper is organized as follows. Section II briefly introduces the PLS and CPLS models. Section III briefly introduces RPLS and proposes both the RCPLS model and the process monitoring techniques. Then, in Section IV, the computational complexity of the model updating for these algorithms is compared through TEP, and the quality-related fault detection performance is compared. The conclusion is given in the last section.

II. PLS AND CPLS MODELS

In a complex industrial process, there are multiple strong correlations for process variable \( X \), resulting in a serious morbidity problem in the ordinary least squares (LS) algorithm for regression applications. To accurately obtain the relationship between \( X \) and \( Y \), S. Wold and C. Albano [2] propose the PLS algorithm.

A. PLS MODEL

The PLS algorithm can extract internal unrelated input latent variables \( t = Xw \) and output latent variables \( u = Yc \) from \( X \) and \( Y \), respectively. \( t \) and \( u \) carry as much variation information as possible for \( X \) and \( Y \), respectively, and their correlation is maximized. In other words, PLS is an algorithm that first obtains the relationship between the latent variables \( t \) and \( u \) and then obtains the relationship between \( X \) and \( Y \). Therefore, the \( Y \)-related information in \( X \) can be accurately extracted and widely used in quality-related process monitoring. Suppose that the input data matrix \( X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n} \) for
n samples and m process variables, and the output data matrix $Y = [y_1, y_2, \ldots, y_p]^T \in \mathbb{R}^{n \times p}$ for n samples and p quality variables. PLS can decompose $X$ and $Y$ into the following format:

$$
\begin{align*}
X &= \hat{X} + X_r = \sum_{i=1}^{\text{rank}(U)} t_i p_i^T + X_r = TP^T + X_r \\
Y &= \hat{Y} + Y_r = \sum_{i=1}^{\text{rank}(U)} b_i q_i^T + Y_r = TBQ^T + Y_r
\end{align*}
$$

(1)

where $T = [t_1, \ldots, t_{\text{rank}(U)}]$ is the scoring matrix; $P = [p_1, \ldots, p_{\text{rank}(U)}]$ and $Q = [q_1, \ldots, q_{\text{rank}(U)}]$ are the load matrices corresponding to $X$ and $Y$, respectively; $X_r$ and $Y_r$ are the residuals; and $B = \text{diag} \{b_1, b_2, \ldots, b_{\text{rank}(U)}\}$ is the inner model regression coefficient diagonal matrix. The number of latent factors A is usually determined by cross-validation. Because $T$ cannot be calculated directly from $X$, a new weight matrix $R = W (P^T W)^{-1}$ is introduced, and $T = XR$ is constructed. From $R$, we can obtain the regression coefficient matrix $C = RBQ^T$. Then, the predictable part of $Y$ can be obtained as $\hat{Y} = XC = XRBQ^T = TBQ^T$.

**B. CPLS MODEL**

The PLS model only monitors $T$ and $X_r$, which means that PLS only monitors the predictable part of $Y$, causing the unpredictable part of $Y$ not to be monitored by the PLS model. The TPLS model is based on PLS and therefore has similar flaws. To this end, Qin [17] proposes the CPLS model. First, the score $U_r$ directly related to $X$ is extracted from the predictable quality $\hat{Y}$, and the weight matrix $R$ of $X$ on $U_r$ is further obtained. Projecting $X$ onto $\text{span}\{R_r\}$ and $\text{span}\{R_p\}$ yields a predictable quality-related subspace $X_r$ and a predictable quality-unrelated but process-related subspace $\hat{X}$. Then, the predictable part of $Y$ is removed from $\hat{Y}$ to obtain the unpredictable quality $\hat{Y}$. Finally, PCA is used to decompose $X_r$ and $\hat{Y}$ into a principal subspace and a residual subspace, respectively. CPLS can decompose $X$ and $Y$ into the following format:

$$
\begin{align*}
X &= U_r R_r^T + T_r P_r^T + \hat{X} \\
Y &= U_r Q_r^T + T_r P_r^T + \hat{Y}
\end{align*}
$$

(2)

The CPLS algorithm process is detailed in [17].

**III. RPLS MODEL**

Most of the PLS literature has used batch processing to model the collected data or update the model. Although batch processing can solve the collinearity problem, it suffers from serious limitations. First, when the amount of data is large, the computational complexity increases as the number of model updates increases, which makes the batch processing algorithm less effective for actual processes. Second, the batch update model often makes repeated use of old data and does not allow the model to reflect current process changes, making it difficult to overcome time-varying disturbances. Although scholars have proposed sliding windows and other methods to discard some old data, the old data are repeatedly used during model updating, which reduces the efficiency of the algorithm. To improve the inefficiency of traditional batch processing algorithms, Helland [26] proposes an RPLS algorithm.

**A. RPLS and BRPLS models**

The RPLS algorithm can update the model without increasing the size of the data matrix. Qin [23] later modified and extended RPLS and proposed the BRPLS algorithm. In [23], Qin not only elaborates on the is listed in TABLE I.

<table>
<thead>
<tr>
<th>TABLE I Modified batch PLS algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm: Modified batch PLS</td>
</tr>
<tr>
<td>Normalize the data matrices $X$ and $Y$. Initialize $i = 0$.</td>
</tr>
<tr>
<td>Step1: Set $i = i + 1$, randomly take a column in $Y$ as $u_i$.</td>
</tr>
<tr>
<td>Step2: Calculate the PLS outer model:</td>
</tr>
<tr>
<td>1) $u_{ii} = u_i$</td>
</tr>
<tr>
<td>2) $w_i = X_i u_i / |u_i| u_i$</td>
</tr>
<tr>
<td>3) $t_i = X w_i / |X w_i|$</td>
</tr>
<tr>
<td>4) $q_i = Y t_i / |Y t_i|$</td>
</tr>
<tr>
<td>5) $u_i = Y q_i$</td>
</tr>
<tr>
<td>If $|u_{ii} - u_i| / |u_{ii}| &lt; 10^{-4}$, go to Step4, else return 1).</td>
</tr>
<tr>
<td>Step4: Calculate loading vector: $p_i = X_i^T t_i$.</td>
</tr>
<tr>
<td>Step5: Calculate inner model regression coefficients: $b_i = u_i^T t_i$.</td>
</tr>
<tr>
<td>Step6: Update $X$ and $Y$:</td>
</tr>
<tr>
<td>$X_i = X_i - t_i p_i^T$</td>
</tr>
<tr>
<td>$Y_i = Y_i - t_i q_i^T$</td>
</tr>
<tr>
<td>Return Step1, until $i = A$.</td>
</tr>
</tbody>
</table>

In [29], it is proved that when the number of latent factors A of PLS is $\text{rank}(X)$, $\|X\| = 0$. Then, by proving $T^T Y_r = 0$ and minimizing the residual $\|Y - XC\|$ in PLS regression coefficients are obtained as follows:

$$
C = RBQ^T = (X^T X)^{-1} X Y
$$

(3)

When new data $(X_{ne}, Y_{ne})$ are obtained, the new PLS model regression coefficients are as follows:

$$
C_{new} = 
\begin{bmatrix}
\left( \begin{bmatrix} X_{ne} X_i \end{bmatrix}^T \right)^{-1} & \left( \begin{bmatrix} Y_{ne} X_i \end{bmatrix}^T \right)
\end{bmatrix} \left( \begin{bmatrix} X_i \end{bmatrix}^T \left( \begin{bmatrix} X_i \end{bmatrix} \right)^T \right)^{-1} \left( \begin{bmatrix} Y_i \end{bmatrix} \right)
\end{bmatrix}
$$

(4)

$T$, which is a unit orthogonal matrix, is obtained by the PLS algorithm of Table I, i.e., $T^T T$ is a unit matrix. A unit orthogonal matrix is an important condition for deriving a recursive algorithm, where the latent factors $A = \text{rank}(X)$. Then, the decomposition of $X$ is as follows:

$$
X = TP^T + X_r = TP^T
$$

(5)
Therefore,
\[
\begin{align*}
X^TX &= PT^TP = PP^T = \left[ P^T \right]^T \left[ P^T \right] \\
X^TY &= PT^TBQ^T + PT^Y = PBQ^T = \left[ P^T \right]^T \left[ BQ^T \right]
\end{align*}
\]
(6)

\[
C_{\text{new}} = \left[ \left( P^T \right)^T \left[ P^T \right], \left[ P^T \right]^T \left[ BQ^T \right] \right]
\]

(7)

Thus, RPLS is an algorithm that uses \( \left[ P^T \right], \left[ BQ^T \right] \) instead of \( \left[ X^T \right], \left[ Y^T \right] \) to update the PLS model. Because the sample lengths of \( P^T \) and \( BQ^T \) are much smaller than the length of the sample data of the old model, RPLS greatly reduces the number of model update calculations and avoids the reuse of old model data. Simultaneously, the RPLS model can be extended to the BRPLS model. Specifically, \( \left[ P^T \right], \left[ BQ^T \right] \) is used instead of \( \left[ X^T \right], \left[ Y^T \right] \) to update the PLS model. The BRPLS recursion process is detailed in [29].

IV. RCPLS MODELS

A. Modified CPLS and RCPLS Models

The use of RPLS for quality-related fault detection has serious drawbacks. Because RPLS is derived based on \( A = \text{rank} \left( X \right) \), \( X \) does not have a residual part. In other words, RPLS cannot produce a quality-related subspace and a quality-unrelated subspace, i.e., RPLS does not have a quality-related fault detection capability. Simultaneously, the RPLS-based RTPLS algorithm cannot monitor the unpredictable part of \( Y \). To effectively update the model and the overall process monitoring, a recursive CPLS (RCPLS) algorithm is proposed in this paper. The number of latent factors \( A \) of RCPLS is determined by cross-validation. The ability to monitor predictable-quality-related faults, predictable-quality-unrelated but process-related faults, and unpredictable-quality-related faults can be achieved for process monitoring.

1) MODIFIED CPLS ALGORITHM

To extend CPLS to RCPLS, we make the following modifications to the CPLS algorithm and show the modifications in Table II: first, \( T_i \) is normalized to unity. Second, we construct a new loading matrix \( P_y = \tilde{Y}_i^T T_i \), where the unpredictable part \( \tilde{Y}_i = Y - U_i Q_i \) can be determined. Although \( \tilde{X}_i \) in TABLE II is unrelated to predictable quality, it is weakly correlated with the unpredictable quality \( \tilde{Y}_i \). Therefore, the following relationship \( \tilde{Y}_i = T_i P_i^T \) must hold because \( T_i \) is a unit orthogonal array, then \( T_i^T T_i = I \). Thus, \( P_i = \tilde{Y}_i^T T_i \).

\[ X \text{ and } Y \text{ can be decomposed into the following format by the modified CPLS algorithm:}
\]
(8)

\[
\begin{align*}
\tilde{X} &= U_i R_i^T + T_i P_i^T \\
\tilde{Y} &= U_i Q_i^T + T_i P_i^T
\end{align*}
\]

TABLE II

<table>
<thead>
<tr>
<th>Algorithm: Modified CPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalize the data matrix ( X ) and ( Y ).</td>
</tr>
<tr>
<td>Step1: ( \text{PLS} \left( X,Y \right) \Rightarrow T, P, W, Q, \cdot \cdot \cdot R = W \left( P^T W \right)^{-1} )</td>
</tr>
<tr>
<td>Step2: Perform SVD on ( Y ) ( \tilde{Y} = U_i D V^T = U_i Q_i^T \cdot\cdot\cdot L = \text{rank} \left( Q \right) \cdot \cdot \cdot Q_i = V D_i \cdot\cdot\cdot R = R_i Q_i^T D_i \cdot\cdot\cdot R_i = R_i )</td>
</tr>
<tr>
<td>Step3: ( U_i \cdot \tilde{Y} V D_i = X R_i \cdot U_i Q_i^T \cdot D_i \cdot R_i = R_i Q_i^T D_i \cdot R_i )</td>
</tr>
<tr>
<td>Step4: ( \tilde{X}_i = \tilde{X} - U_i R_i^T \cdot R_i^T \left( R_i^T \right)^{-1} R_i )</td>
</tr>
<tr>
<td>Step5: ( L = \text{rank} \left( \tilde{X}_i \right) \cdot\cdot\cdot \text{perform PCA on} \cdot\cdot\cdot \text{with} \cdot\cdot\cdot \text{elements} \cdot\cdot\cdot \text{Scale} \cdot\cdot\cdot \text{to ensure that} \cdot\cdot\cdot \text{norm} )</td>
</tr>
<tr>
<td>Step6: ( \tilde{Y}_i = T_i P_i^T, T_i^T T_i = I \cdot\cdot\cdot \cdot\cdot \cdot\cdot )</td>
</tr>
</tbody>
</table>

2) RCPLS ALGORITHM

In this section, we propose the RCPLS algorithm based on the modified CPLS algorithm. \( \left( U_i R_i^T \right) \) is the predictable-quality-related and unpredictable-quality-unrelated part, \( T_i P_i^T \) is the unpredictable-quality-related and predictable-quality-unrelated part, \( U_i Q_i^T \) is the predictable quality part, and \( T_i P_i^T \) is the unpredictable quality part. Thus, \( U_i Q_i^T \) is unrelated to \( T_i P_i^T \) and \( U_i R_i^T \) is unrelated to \( T_i P_i^T \) [17]. Therefore, the covariance of \( X \) and the cross-correlation between \( X \) and \( Y \) are given as follows:

\[
\begin{align*}
X^TX &= \left( R_i^T \right)^T R_i^T + P_i P_i^T = \left( R_i^T \right)^T P_i P_i^T \\
X^TY &= \left( R_i^T \right)^T Q_i^T + P_i P_i^T = \left( R_i^T \right)^T Q_i^T P_i P_i^T \\
X^TY &= \left( R_i^T \right)^T \left( R_i^T \right)^T + P_i P_i^T = \left( R_i^T \right)^T \left( R_i^T \right)^T + P_i P_i^T
\end{align*}
\]

(9)

When new data \( \{ X_n, Y_n \} \) are obtained, it is known from (11) that

\[
C_{\text{new}} = \begin{bmatrix}
\left( R_i^T \right)^T & \left( R_i^T \right)^T & \left( R_i^T \right)^T & \left( R_i^T \right)^T & \left( R_i^T \right)^T
\end{bmatrix}
\begin{bmatrix}
P_i^T \\
P_i^T \\
P_i^T \\
P_i^T \\
P_i^T
\end{bmatrix}
\begin{bmatrix}
X_n \\
X_n \\
X_n \\
X_n \\
Y_n
\end{bmatrix}
\]

(10)
Then, the RCPLS algorithm uses
\[
\begin{bmatrix}
R'_{t} \\
P^T_{t} \\
X_{n} \\
Y_{n}
\end{bmatrix}
\]
instead of
\[
\begin{bmatrix}
X \\
X_{n} \\
Y \\
Y_{n}
\end{bmatrix}
\]
to update the modified CPLS model. The corresponding block RCPLS (BRCP) algorithm updates the modified CPLS model with
\[
\begin{bmatrix}
R'_{t} \\
P^T_{t} \\
R^T_{nc} \\
P^T_{nc}
\end{bmatrix}
\]\ninstead of
\[
\begin{bmatrix}
X \\
X_{n} \\
Y \\
Y_{n}
\end{bmatrix}
\]. In the monitoring of slow time-varying industrial processes, constructing the modified CPLS model from \(\{X_{n}, Y_{n}\}\), updating the current model parameter matrix
\[
\begin{bmatrix}
R'_{t} \\
P^T_{t} \\
Q^T \\
P^T_{y}
\end{bmatrix}
\]
and the new model parameter matrix
\[
\begin{bmatrix}
R'_{nc} \\
P^T_{nc} \\
Q^T_{nc} \\
P^T_{ny}
\end{bmatrix}
\]
to a model through the BRCPLS algorithm do not improve the updating efficiency of the monitoring model but rather slightly increases the computational burden. Therefore, this article only uses the RCPLS algorithm for process monitoring. RCPLS can effectively reduce the number of model update calculations while also enhance the process tracking ability of the model.

**B. PROCESS MONITORING TECHNOLOGY**

The TEP simulation laboratory was proposed by Downs and Vogel of the TE Chemical Company in the United States in 1993. The department designed a research platform for monitoring, diagnosing and optimizing processes, constructing the modified CPLS model from \(\{X_{n}, Y_{n}\}\),

the predictable-quality-related subspace score \(u_{c}\),

the predictable-quality-unrelated subspace score \(t_{c}\),

and the unpredictable quality subspace score \(t_{y}\)
can be obtained.

Then, the predictable-quality-related subspace score \(u_{c}\),

the predictable-quality-unrelated subspace score \(t_{c}\),

and the unpredictable quality subspace score \(t_{y}\)
can be constructed from the sample data \(\{x, y\}\) as follows:

\[
\begin{align*}
\hat{t}_{c} &= x - R'_{c}u_{c} \\
\hat{y}_{c} &= y - Q_{c}u_{c}
\end{align*}
\]

where \(\hat{t}_{c} = x - R'_{c}u_{c}\), \(\hat{y}_{c} = y - Q_{c}u_{c}\).

The monitoring statistics [17] of the corresponding subspace can be constructed from the scores \(u_{c}\), \(t_{c}\), and \(t_{y}\) as follows:

\[
\begin{align*}
\hat{T}_{c}^2 &= (N - 1)u_{c}^T u_{c} \\
\hat{T}_{y}^2 &= t_{y}^T A^T t_{y} \\
\hat{T}_{y}^2 &= t_{y}^T A^T t_{y}
\end{align*}
\]

where \(A = T_{c}' T_{c}/(N - 1)\) and \(N\) is the number of current model data samples.

To facilitate the update of the RCPLS model control limits, the control limits of \(T_{c}^2\), \(T_{y}^2\), and \(\hat{T}_{y}^2\) are structured as follows:

\[
\begin{align*}
J_{th,c}^2 &= g_{th,c}^2, \quad g_{c} = S_{c}/\mu_{c}, \quad h_{c} = 2\mu_{c}^2/S_{c}, \\
J_{th,y}^2 &= g_{th,y}^2, \quad g_{y} = S_{y}/\mu_{y}, \quad h_{y} = 2\mu_{y}^2/S_{y}
\end{align*}
\]

In \(\{S_{c}, \mu_{c}\}_{c}, \{S_{y}, \mu_{y}\}_{y}\), \(S\) and \(\mu\) are the sample mean and variance of the corresponding model data statistics. \(\chi_{h,a}^2\) is the \(\chi^2\) distribution threshold with degree of freedom \(h\) and confidence \(\alpha\).

During process monitoring, after measuring new sample data \(\{x_{new}, y_{new}\}\), new scores \(u_{new}, t_{new}\), and \(\hat{t}_{new}\) can be calculated. New statistics \(\hat{T}_{c}^2_{new}\), \(\hat{T}_{y}^2_{new}\), and \(\hat{T}_{y}^2_{new}\) are calculated from the new score, and then, the new statistics are compared with the corresponding model control limits as follows:

1. If \(\hat{T}_{c}^2_{new} > J_{th,c}^2\), there is a predictable-quality-related fault in \(x_{new}\).
2. If \(\hat{T}_{y}^2_{new} > J_{th,y}^2\), there is a predictable-quality-unrelated but process-related fault in \(x_{new}\).
3. If \(\hat{T}_{y}^2_{new} > J_{th,y}^2\), there is an unpredictable-quality-related fault.

2) PROCESS MONITORING TECHNOLOGY OF RCPLS

In this section, a set of RCPLS process monitoring technologies, listed in Table III, is presented based on the RCPLS model and its process monitoring indicators. \(X_{c}\) and \(Y_{c}\) are defined as the model data storage matrix, and we let \(\{X_{n}, Y_{n}\}\) have a maximum length parameter \(MaxL\). We use the model update parameters \(WL\) to decide whether to update the model. When the number of new model data samples has accumulated up to \(WL\), the RCPLS model will be updated, and monitoring continues. \(T_{c}^2_{new}\), \(T_{y}^2_{new}\), and \(\hat{T}_{y}^2_{new}\) are defined as test data statistic storage vectors, and \(J_{c}^2_{c,new}\), \(J_{y}^2_{y,new}\), and \(J_{y}^2_{y,new}\) are defined as control limit storage vectors for the corresponding statistics.

The RCPLS process monitoring technology can use model parameters and new data to quickly update the monitoring model, greatly reducing the number of model update calculations when implementing the classification alarm function. The updated model of the RCPLS monitoring technology is more efficient than the traditional batch update model. By abandoning the old data, the use of the
current model parameters and the new data model update method effectively improves the process model tracking ability of the monitoring model. The detailed steps of RCPLS process monitoring are shown in TABLE III.

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>RCPLS Process Monitoring Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized the data matrix $X$ and $Y$, initialize $X_j, Y_j, T^2_{new}, T^2_{out}$, and $T^2_{out}$ to be empty, and let $i = 0$.</td>
<td>Step1: Perform modified CPLS model on X and Y to obtain $T_j, R_j, R_j^{T}, Q_j, P_j$, and $P_j$. Calculate the model statistics $T^2_{new}, T^2_{out}$, and the control limits $J_{a,j}, J_{a,j}, J_{a,j}$.</td>
</tr>
<tr>
<td>For each online test sample $[X_{new}, Y_{new}]$, calculate statistics $T^2_{new}, T^2_{out}$ by (13) and store them: $T^2_{new} = [T^2_{new}, T^2_{new}]$ and $T^2_{out} = [T^2_{out}, T^2_{out}]$.</td>
<td>Step2:</td>
</tr>
<tr>
<td>Fault detection: If each statistic is below the control limit, store the current sample: $X_j = [X_j, X_{new}], Y_j = [Y_j, Y_{new}], i = i + 1$.</td>
<td>1. If $T^2_{new} &gt; J_{a,j}$, a predictable-quality-related fault has occurred in $X_{new}$.</td>
</tr>
<tr>
<td>2. If $T^2_{out} &gt; J_{a,j}$, a predictable-quality-unrelated but process related fault has occurred in $X_{new}$.</td>
<td>3. If $J_{a,j} &gt; J_{a,j}$, a unpredictable-quality-related fault has occurred.</td>
</tr>
<tr>
<td>If $i = WL$, Formulate $[R_j^{T}, Q_j, P_j^{T}, P_j]$ and run modified CPLS on it. Update control limits, $J_{a,j} = J_{a,j}, J_{a,j} = J_{a,j}$, and $J_{a,j} = J_{a,j}$. If the sample number of ${X_j, Y_j}$ exceeds $MaxL$, remove redundant old data. Set $i = 0$, and return to Step 2.</td>
<td>Step3:</td>
</tr>
</tbody>
</table>

3) MODEL UPDATING
The RCPLS model update matrix consists of two parts, as shown in Eqn. 11. One part consists of normal test data $(X_j, Y_j)$, and the other part consists of model parameters.

Normal data part: In the fault detection process, the principle of updating test data as a model is that the test sample is stored in the model update matrix without any faults after detection.

Model parameters part: historical model parameters are calculated from the previous modeling data $\{X, Y\}$ through Table II. The model parameters $R_j^{T}, Q_j, P_j, P_j$, and $P_j$, are calculated. Eqn. 11 shows that $[R_j^{T}, Q_j, P_j^{T}, P_j]$ replaces the original data; thus, the update matrix of the model can be established as $[X_n, Y_n]$ instead of $[X_n, Y_n]$. The data matrix in the RCPLS model is replaced by the original process data by using parameters, the dimensionality is reduced, and the computational burden is effectively reduced.

V. NUMERICAL EXAMPLE AND CASE STUDY
A. NUMERICAL EXAMPLE
In this section, we use synthetic simulations to create a number of representative fault scenarios to demonstrate the effectiveness of CPLS in terms of detecting quality-related and input-relevant faults. The advantages of CPLS-based monitoring over other existing methods are highlighted using simulation cases.

The simulated numerical example without faults is as follows.

$$\begin{cases}
 x_k = A z_k + e_k \\
 y_k = c x_k + v_k
\end{cases} \quad (16)$$

where $z_k \in \mathbb{R}^3$, $z_k, i \sim U([0,1])$, $v_k \sim N(0,0.1)$.

$$C = \begin{bmatrix}
 2 & 2 & 1 & 1 & 1 \\
 3 & 1 & 0 & 4 & 0
\end{bmatrix}, \quad A = \begin{bmatrix}
 1 & 3 & 4 & 4 & 0 \\
 3 & 0 & 1 & 4 & 1 \\
 1 & 1 & 3 & 0 & 0
\end{bmatrix}, \quad e_k \in \mathbb{R}^5$$

$e_i \sim N(0,0.2^2)(i = 1,..,5)$. $U([0,1])$ denotes the uniform distribution in the interval $[0,1]$.

A fault is added in the following form in the input space,

$$x_k = x_k^* + \Xi_x f_x \quad (17)$$

or in the output space

$$y_k = y_k^* + \Xi_y f_y \quad (18)$$

where $x_k^*$ and $y_k^*$ are the fault-free values generated from (16), $\Xi_x$ and $\Xi_y$ are the fault-free values, respectively, and $f_x$ and $f_y$ are the fault magnitudes, respectively.

First, 100 samples are generated under normal operating conditions to build the regression model. Another 2000 samples are generated for detection purposes, in which the first 1800 samples are fault-free samples, while the last 200 samples are faulty samples. The PLS factor number $l = 4$ is determined by 10-fold cross-validation.

B. FAULT OCCURS IN CVS ONLY
To generate a fault that occurs in the covariation space only, we choose $\Xi_y$ to be the first column of $R_y$ and normalize it to unit norm; thus, the fault occurs in CVS only. The fault detection indices are shown in Figure 1. The result indicates that only $T^2_c$ detects the fault, whereas other fault detection indices are not affected by the fault. This result implies that
the fault detected in the input space by $T^2_e$ is an output-relevant fault.

**C. FAULT OCCURS IN IPS ONLY**

Let $X_1$ be the first column of $P$, and thus, the fault occurs in IPS only. The fault detection indices are shown in Figure 2. This fault does not affect output $y$. The result indicates that $T^2_e$ detects the input-relevant fault; however, it is not output relevant.

**D. FAULT OCCURS IN OPS ONLY**

Let $X_1$ be the first column of $f$; thus, the fault occurs in OPS only. To see if this fault will affect the output quality, we calculate the fault detection indices, as shown in Figure 3. The result indicates that although the fault occurs only in the residual space of the input data, $T^2_{new}$ detects the fault as output relevant and is unpredictable from the input.

**E. FAULT DETECTION DELAY**

The delayed detection of faults occurring in CVS, IPS, and OPS spaces results in a fault detection delay of 0, indicating that RCPLS can detect the fault immediately in the numerical simulation when a fault occurs. The specific monitoring results can be seen in the spatial monitoring of the faults in Figs. 1, 2, and 3.

**VI. TENNESSEE-EASTMAN PROCESS SIMULATION APPLICATION**

The Tennessee Eastman process (TEP) is a simulation model developed by Prof. Richard Bratz for a large system research laboratory studying the process control and detection of industrial chemical processes. TEP can be used for not only the research and evaluation of process control technology but also a variety of research areas, including plant-wide control and multivariable control issues. The TEP includes 41 measurement variables and 12 control variables, and the process is usually sampled every 3 minutes. Gaussian noise is included in all process measurements. A detailed description of the TEP can be found in [31]. The TEP simulation data can be downloaded from Prof. Richard Bratz's website. According to the standard introduction in [15], 9 faults [IDV(1, 2, 5-8, 10, 12, 13)] are considered to be quality-related faults, and 5 faults [IDV(3, 4, 9, 11, 15)] are considered to be quality-unrelated faults. Based on the TEP simulation data, the online monitoring of quality-related faults using the CPLS model and RCPLS model is performed. In online monitoring, the RCPLS model update efficiency is verified by comparing the number of RCPLS and CPLS model update calculations. Through quality-related and process-related fault detection experiments, RCPLS is verified to have quality-related and process-related fault detection capabilities. Then, a comparison of RCPLS and CPLS shows that RCPLS has stronger quality-related fault-detection capabilities.

**A. PARAMETER INITIALIZATION**

The input variables $X \in \mathbb{R}^{n \times m}$ for this experiment are XMEAS (1-36) and XMV (1-11), and the output variables $Y \in \mathbb{R}^{m \times p}$ are XMEAS (37-41), where XMEAS (1-36) are the process measurement variables, XMV (1-11) are the manipulated variables, and XMEAS (37-41) are the quality measurement variables. The initial modeling data are the d00 dataset with a sample size of 500. The number of online test data samples is 4480, of which the first 4000 are normal data and the last 480 are fault data. The number of samples with 480 faults is derived from 9 quality-related
fault datasets (d01, d02, d05-08, d10, d12, and d13), i.e., IDV (1, 2, 5-8, 10, 12, 13). For RCPLS and CPLS, the latent parameter $A = 4$ is obtained via 10-fold cross-validation.

B. COMPARISON OF CALCULATIONS

To more clearly compare the computational complexity of the model updating, we compare the computational complexity of CPLS and RCPLS by the method of [32] and the detail distribution is list in the TABLE IV.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCPLS</td>
<td>cost of PLS on a $N \times S$ matrix + 1 SVDs on an $m \times m$ matrix + 1 SVD on a $l \times l$ matrix</td>
</tr>
<tr>
<td>CPLS</td>
<td>cost of PLS on a $n \times S$ matrix + 1 SVDs on an $m \times m$ matrix + 1 SVD on a $l \times l$ matrix</td>
</tr>
</tbody>
</table>

In TABLE IV, it is clear that the difference in computational complexity between the two algorithms in model updating lies in the dimensionality of the matrix in the PLS model. Thus, we only need to compare the dimensionality of the model update matrix to judge the computational complexity of the two algorithms.

This section is based on IDV (1) online detection data with a sample size of 4480. The online monitoring process of RCPLS and CPLS is performed in parallel. During the monitoring process, the model is updated several times according to the RCPLS process monitoring technology. According to the RCPLS model update time, the CPLS model is updated simultaneously.

![Model updating time](image)

**FIGURE 4.** Process monitoring results for IDV(1) by RCPLS.
Fig. 4 shows the RCPLS process monitoring of IDV(1). As seen from the partial enlargement of the statistical section in Fig. 4, the RCPLS model is updated seven times. The update moments are 508, 1025, 1571, 2058, 2573, 3101, and 3621. Simultaneously, traditional batch model update operations are performed on CPLS at those seven moments. The initial model data sample number is 500. The number of newly added model data samples is 500 in each update. After modeling, 600 new data samples are retained, and except old data are forgotten in \( \{X, Y\} \). Assume that the number of data samples is calculated. The numbers of calculations for RCPLS and CPLS are listed in Table V.

Table V shows that the number of model update calculations of RCPLS is smaller than that of CPLS. RCPLS requires 10171 fewer calculations than CPLS. Because RCPLS uses model parameters
\[
\begin{bmatrix} R_i^T & Q_i^T \\ P_i^T & P_i^T \end{bmatrix} \in \mathbb{R}^{(47+5)T} \]
and updateable data \( \{X, Y\} \in \mathbb{R}^{47+5} \) to update the model, CPLS uses the old data \( \{X_{\text{old}}, Y_{\text{old}}\} \in \mathbb{R}^{500(47+5)} \) and new data \( \{X_{\text{new}}, Y_{\text{new}}\} \in \mathbb{R}^{100(47+5)} \) to update the model, where \( up = \begin{cases} 500, & j = 1, 2 \\ 1100, & j \geq 3 \end{cases} \), in which \( j \) is the number of model updates. With increasing number of updates, the computational cost of the CPLS model continues to increase; however, the proposed RCPLS model continues to require 1147 calculations. It can be seen that RCPLS greatly reduces the number of calculations and effectively improves both the efficiency of model updating and the quality-related and process-related fault detection performance.

### C. FAULT DETECTION

In this section, process monitoring based on RCPLS and CPLS is performed for quality-related fault IDV (1,2,5,8,10,12,13) and quality-unrelated fault IDV (3, 4, 8, 11, 15). From the TABLE VI, it can be seen that the RCPLS detection performance in IDV (1,8) is only slightly lower than that of CPLS by 0.41 and 1.24 percentage points. In IDV (2,7,10,12,13), the FDR of RCPLS is higher than CPLS. In general, RCPLS has a better effectiveness of monitoring for quality related faults.

In TABLE VII, it is easy to conclude that the false alarm rate of RMPLS is higher than that of CPLS when monitoring for quality-unrelated faults. In the monitoring fault IDV (3, 4, 9, 15), the false alarm rate of RCPLS and CPLS is low; they can effectively identify the quality-independent fault and reduce the false alarm rate. In the monitoring of fault IDV (11), RCPLS suffers from a high false alarm rate. When monitoring such faults, false alarms are readily produced by the system. In general, the RCPLS structure leads to an increase in the false positive rate, which may lead to a decrease in system stability.
D. QUALITY-RELATED AND PROCESS-RELATED FAULT DETECTION CAPABILITIES

To verify the quality-related and process-related fault detection performance of RCPLS, the RCPLS monitoring results for the quality-related fault IDV (5) are given, as shown in Fig. 5. In Fig. 5, the fault is detected at sample 4000. However, the fault of the $T^2_{cw}$ and $T^2_{ynew}$ detection at sample 4200 eventually disappears, and $T^2_{ynew}$ can continue to detect a slight deviation of the condenser cooling water flow. The fault that occurs in samples 4000 to 4200 in the online detection data IDV (5) is related to quality. After 4200, the quality-related fault disappears; however, a slight deviation of the condenser cooling water flow remains. $T^2_{ynew}$ succeeds in detecting small deviations that are not
related to quality but that are related to the process. The above shows that RCPLS can detect quality-related and process-related faults.

To verify that RCPLS can discriminate between quality-related faults and process-related faults, the RCPLS monitoring results for the quality-unrelated fault IDV (4) are given, as shown in Fig. 6. In Fig. 6, it can be seen that $T^2_{\text{new}}$ and $T^2_{\text{new}}$ have almost no faults detected, and the $T^2_{\text{new}}$ statistic FDR is as high as 100%; this means that the fault is not related to predictable or unpredictable quality and is only related to the process. Therefore, RCPLS can successfully identify IDV (4) as a quality-unrelated but process-related fault.

E. Fault detection delay time

The effectiveness of the fault detection is important for process monitoring. In this section, we separately record the lag time of the fault occurrence for the CPLS and RCPLS models in TEP monitoring, where the sampling time represents each time point. The results are shown in TABLE VII.

<table>
<thead>
<tr>
<th>Fault number</th>
<th>CPLS (A=4)</th>
<th>RCPLS (A=4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T^2_{\text{new}}$</td>
<td>$T^2_{\text{new}}$</td>
</tr>
<tr>
<td>IDV(1)</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>IDV(2)</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>IDV(5)</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>IDV(6)</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>IDV(7)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>IDV(8)</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>IDV(10)</td>
<td>94</td>
<td>96</td>
</tr>
<tr>
<td>IDV(12)</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>IDV(13)</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

The detection delay of the $T^2_{\text{new}}$ and $T^2_{\text{new}}$ spaces is analyzed separately. In $T^2_{\text{new}}$, the hysteresis value of CPLS for each type of fault monitoring is close to that of RCPLS, and the monitoring of IDV (10) lags behind. In $T^2_{\text{new}}$, the fault monitoring hysteresis of RCPLS is generally lower than that of CPLS. RCPLS can provide an instant alarm for IDV (1, 5, 6, 7), CPLS suffers a large lag for IDV(10), and RCPLS reduces the lag value for the fault from 96 sampling times to 19 sampling times, thereby further improving the effectiveness of the alarm.

VII. CONCLUSIONS

In this paper, an RCPLS model is proposed based on the fact that the traditional batch processing monitoring model has large computational complexity when applied to complex and slow time-varying industrial processes. The model uses historical model parameters and new data to achieve more efficient model updates. Based on the TEP, the model update calculations of the RCPLS and CPLS are compared. The results show that the RCPLS model update requires fewer calculations than CPLS. RCPLS achieves a high quality-related and process-related fault detection performance and is superior to CPLS for quality-related fault detection. In addition, RCPLS can accurately detect both predictable quality-related faults and unpredictable quality-related faults. This article also provides a set of RCPLS process monitoring technologies that can effectively reduce the number of model update calculations and filter modellable data to achieve an adaptive model update function. The RCPLS process monitoring technology can effectively reduce equipment maintenance costs through the real-time monitoring of slow time-varying industrial processes. However, note that although the fault detection rate of RCPLS is improved, the false positive rate is increased, which may cause false alarms in industrial processes. Therefore, future work could attempt to decrease the fault alarm rate through pre-treatment.

REFERENCES


ZHONGYING XU received the B.S. and M.S. degrees from the Xi’an Research Institute of High Technology, Xi’an, China.

Currently, he is pursuing the Ph.D. degree in control science and engineering. His research interests include stochastic system analysis, nonlinear system modeling and its application, complex system fault diagnosis.

XIANGYU KONG received the bachelor’s degree from the Beijing Institute of Technology, Beijing, China, in 1990, the master’s degree from Xi’an Research Institute of High Technology, Xi’an, China, in 2000, and the Ph.D. degree in control science and engineering from the Xi’an Jiao tong University, Xi’an, in 2005.

Currently, he is a professor with the Xi’an Research Institute of High Technology. His research interests include adaptive signal processing, system modeling, and fault diagnosis.

JIAYU LUO was born in Xichang, Sichuan, China, in 1994. received the B.S. degree from Hunan University, Hunan, China, in 2017.

Currently, he is pursuing the M.S. degree in control science and engineering with the Xi’an Research Institute of High Technology. His research interests include feature extraction and process monitoring.

CHANGHUA HU received the Bachelor’s and Master’s degrees from the Rocket Force University of Engineering, Xi’an, China, in 1987 and 1990, respectively, and the Ph.D. degree from the Northwestern Polytechnical University, Xi’an, in 1996.

Currently, he is a Professor with the Rocket Force University of Engineering. His current research has been supported by the National Science Fund for Distinguished Young Scholars of China. His research interests include fault diagnosis and prediction, life prognosis, and fault tolerant control.