processes

MDPI

Article

A Novel Dynamic Process Monitoring Algorithm: Dynamic Orthonormal Subspace Analysis

Weichen Hao [1], Shan Lu [2], Zhijiang Lou [2,*], Yonghui Wang [3], Xin Jin [1] and Syamsunur Deprizon [3,4]

[1] School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113005, China; weichenhenry@163.com (W.H.); jinxin@lnpu.edu.cn (X.J.)

[2] Institute of Intelligence Science and Engineering, Shenzhen Polytechnic, Shenzhen 518055, China; lushan@szpt.edu.cn

[3] Faculty of Engineering, Technology & Built Environment, UCSI University, Kuala Lumpur 56000, Malaysia; 1002164990@ucsiuniversity.edu.my (Y.W.); deprizon@ucsiuniversity.edu.my (S.D.)

[4] Postgraduate Department, Universitas Bina Darma, Palembang 30111, Indonesia

* Correspondence: louzhijiang@szpt.edu.cn; Tel.: +86-15001009604

Abstract: Orthonormal subspace analysis (OSA) is proposed for handling the subspace decomposition issue and the principal component selection issue in traditional key performance indicator (KPI)-related process monitoring methods such as partial least squares (PLS) and canonical correlation analysis (CCA). However, it is not appropriate to apply the static OSA algorithm to a dynamic process since OSA pays no attention to the auto-correlation relationships in variables. Therefore, a novel dynamic OSA (DOSA) algorithm is proposed to capture the auto-correlative behavior of process variables on the basis of monitoring KPIs accurately. This study also discusses whether it is necessary to expand the dimension of both the process variables matrix and the KPI matrix in DOSA. The test results in a mathematical model and the Tennessee Eastman (TE) process show that DOSA can address the dynamic issue and retain the advantages of OSA.

Keywords: process monitoring; key performance indicators; orthonormal subspace analysis; dynamic process

check for updates

Citation: Hao, W.; Lu, S.; Lou, Z.; Wang, Y.; Jin, X.; Deprizon, S. A Novel Dynamic Process Monitoring Algorithm: Dynamic Orthonormal Subspace Analysis. *Processes* **2023**, *11*, 1935. https://doi.org/10.3390/pr11071935

Academic Editor: Wei Sun

Received: 21 April 2023
Revised: 1 June 2023
Accepted: 8 June 2023
Published: 27 June 2023

1. Introduction

Process monitoring and fault detection are two important aspects of process systems engineering because they are the key issues to address in order to ensure the safety and the normal operation of industrial processes [1].As such, traditional data-driven algorithms such as principal components analysis (PCA) [2] and independent components analysis (ICA) [3] have been proposed to monitor processes and to improve the product quality. PCA and ICA can effectively detect faults in a process. However, in the actual production process at a modern industrial plant, there are a large number of controllers, sensors and actuators distributed widely, and not all data need to be analyzed [4,5]. That is to say, not all process variables directly affect the safety and the product quality. The information highly relevant to the product quality and economic benefits are called key performance indicators (KPIs), and their role should be emphasized in process monitoring [6,7]. It is worth mentioning that both PCA and ICA monitor KPI-related and KPI-unrelated components simultaneously, and they perform poorly in detecting faults in KPI-related components because the fault information might be submersed in the disturbances of numerous KPI-unrelated components. As such, KPI-related process monitoring such as partial least squares (PLS) [8] and canonical correlation analysis (CCA) [9] algorithms have developed rapidly in recent decades, and this development is essential for ensuring production safety and obtaining superior operation performance.

However, there are still some drawbacks to these traditional KPI algorithms. First, the residual subspace calculated by the PLS algorithm is non-orthogonal to the principal components (PCs) subspace, which means that some KPI-related information may leak into the residual spaces [10,11]. Second, the CCA algorithm requires KPIs to be available during both offline training and online monitoring stages as it uses KPI variables to construct indices [12,13]. Third, both PLS and CCA algorithms are unable to extract PCs [14,15].

To address the above issues in traditional KPI-related algorithms, Lou et al. proposed orthonormal subspace analysis (OSA) [16]. OSA can divide the process data and KPIs into three orthonormal subspaces, namely, subspaces of KPI-related components, KPI-unrelated components in process data, and process-unrelated components in KPIs. Furthermore, the cumulative percent variance method is used to select the number of PCs in an OSA algorithm. Due to the ability of the OSA algorithm to independently monitor each subspace, the OSA algorithm is not limited by KPIs during the offline and online stages.

The original OSA was proposed for addressing the monitoring issues in static process problems, so it assumes that the observations are time-independent. However, dynamic features widely exist in most industrial processes, and, hence, the auto-correlation relationships in variables interfere with the extraction of the KPI-related information [17,18]. Therefore, the subspaces obtained by the OSA algorithm are not orthonormal in dynamic processes.

The "time lag shift" method, which lists the historical data as additional variables to the original variable set, is an effective measure for handling the dynamic issue, and it has been adopted in the PLS and CCA algorithms, i.e., the dynamic PLS (DPLS) and dynamic CCA (DCCA) algorithms. Therefore, in this paper, the "time lag shift" method is also combined with the OSA algorithm, named the dynamic OSA (DOSA) algorithm, and is applied to the Tennessee Eastman (TE) process to illustrate its efficiency.

The contributions of this study are as follows. First, this study proposes DOSA for dealing with the low detection rate problem caused by the dynamics processes. DOSA can determine whether the fault in a dynamics process originates from KPI-related or KPI-unrelated process variables or the measurement of KPIs. Second, this study discusses whether it is necessary to expand the dimension of both the process variables matrix and the KPI matrix in order to reduce the computation. At the same time, a new method to select the time lag number in the "time lag shift" structure is proposed. Additionally, we analyze the impact of the sampling period on DOSA. Third, we place an emphasis on the real-time nature of information and design new monitoring indices. Finally, this study compares the detection rates of the OSA, DOSA, DPLS, and DCCA algorithms.

The remainder of this paper is organized into five sections. Section 2 discusses the classical OSA algorithm and the "time lag shift" method. Section 3 proposes DOSA for dynamics process monitoring. Section 4 compares the DOSA algorithm with other KPI-related algorithms based on TE process testing. Section 5 reviews the contributions of this work.

2. Methods

2.1. Orthonormal Subspace Analysis

Here, we take $\boldsymbol{X} \in \boldsymbol{R}^{n\times s}$ as the process variables matrix (where n is the number of samples, and s is the number of process variables), and the standard PLS identification technique introduces the KPI matrix as $\boldsymbol{Y} \in \boldsymbol{R}^{n\times r}$ (where r is the number of KPIs). OSA decomposes both $\boldsymbol{X}$ and $\boldsymbol{Y}$ into the following bilinear terms:

$$\begin{cases} \boldsymbol{X} = \boldsymbol{T}_{com}\Xi_{\boldsymbol{X}}^{T} + \boldsymbol{E}_{OSA} \\ \boldsymbol{Y} = \boldsymbol{T}_{com}\Xi_{\boldsymbol{Y}}^{T} + \boldsymbol{F}_{OSA} \end{cases}, \tag{1}$$

where $\boldsymbol{T}_{com} \in \boldsymbol{R}^{n\times\phi}$ (ϕ is the number of principal components) is the common latent variables shared by $\boldsymbol{X}$ and $\boldsymbol{Y}$; $\Xi_{\boldsymbol{X}} \in \boldsymbol{R}^{s\times\phi}$ and $\Xi_{\boldsymbol{Y}} \in \boldsymbol{R}^{r\times\phi}$ are the transformation matrices; and $\boldsymbol{E}_{OSA} \in \boldsymbol{R}^{n\times s}$ and $\boldsymbol{F}_{OSA} \in \boldsymbol{R}^{n\times r}$ are the residual matrices.

Then, OSA, along with PLS and CCA, is called 'KPI-related algorithm'. As opposed to PLS and CCA, the extracted subspaces of OSA are proved to be orthogonal [16]. That is to say, $\boldsymbol{T}_{com}$, $\boldsymbol{E}_{OSA}$, and $\boldsymbol{F}_{OSA}$ are orthogonal in Equation (1), and, most importantly, they can be monitored independently.

2.2. The "Time Lag Shift" Method

The proposed OSA algorithm in Section 2.1 implicitly assumes that the current observations are statistically independent to the historical observations [19,20]. That is to say, OSA only considers the correlation between variables at the same time but does not consider the mutual influence of variables at different times. However, most data from industrial processes show degrees of dynamic characteristics; that is, the sampling data at different times are correlated. For such a process, the static OSA algorithm is not applicable.

The most common method to address such a problem is to use an autoregressive (AR) model to describe the dynamic characteristics. Similarly, the OSA algorithm can be extended to take into account the serial correlations by augmenting each observation vector, $\boldsymbol{X}(t) \in \boldsymbol{R}^{1\times s}$ or $\boldsymbol{Y}(t) \in \boldsymbol{R}^{1\times r}$, at the current time t with the previous l_x or l_y observations in the following manner [21]:

$$\begin{cases} \widetilde{\boldsymbol{X}}(t) = [\boldsymbol{X}(t), \boldsymbol{X}(t-1), \ldots, \boldsymbol{X}(t-l_x)] \in \boldsymbol{R}^{1\times[(l_x+1)\times s]} \\ \widetilde{\boldsymbol{Y}}(t) = \left[\boldsymbol{Y}(t), \boldsymbol{Y}(t-1), \ldots, \boldsymbol{Y}(t-l_y)\right] \in \boldsymbol{R}^{1\times[(l_y+1)\times s]} \end{cases}, \tag{2}$$

As known in Equation (2), the first s columns of $\widetilde{\boldsymbol{X}}(t)$ and the first r columns of $\widetilde{\boldsymbol{Y}}(t)$ represent the data at the current time, and the rest represent the data at the past time. For n sampling times, one can obtain the augmented matrices $\widetilde{\boldsymbol{X}} \in \boldsymbol{R}^{n\times[(l_x+1)\times s]}$ and $\widetilde{\boldsymbol{Y}} \in \boldsymbol{R}^{n\times[(l_y+1)\times s]}$.

By performing dimension expansion on the data matrix in Equation (2), the static OSA methods can be used to analyze the autocorrelation, cross-correlation, and hysteresis correlation among the data synchronously. That is to say, $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{Y}}$ will be decomposed by OSA. More details can be found in Section 3.

3. Dynamics Orthonormal Subspace Analysis

3.1. Determination of the Lag Number

As the traditional lag determination methods, such as the Akaike information criterion (AIC) [22] and the Bayesian information criterion (BIC) [23], are only suitable for a steady state, a new lag determination method should be proposed for DOSA.

Suppose the relationship between the data at the current time and the past time is as follows:

$$\begin{cases} \boldsymbol{X}(t) = \boldsymbol{X}(t-1)\boldsymbol{A}_1 + \boldsymbol{X}(t-2)\boldsymbol{A}_2 + \ldots + \boldsymbol{X}(t-l_x)\boldsymbol{A}_{l_x} + \boldsymbol{D}_x(t) = \overline{\boldsymbol{X}}(t)\overline{\boldsymbol{A}} + \boldsymbol{D}_x(t) \\ \boldsymbol{Y}(t) = \boldsymbol{Y}(t-1)\boldsymbol{B}_1 + \boldsymbol{Y}(t-2)\boldsymbol{B}_2 + \ldots + \boldsymbol{Y}(t-l_y)\boldsymbol{B}_{l_y} + \boldsymbol{D}_y(t) = \overline{\boldsymbol{Y}}(t)\overline{\boldsymbol{B}} + \boldsymbol{D}_y(t) \end{cases}, \tag{3}$$

where $\overline{\boldsymbol{X}}(t) = [\boldsymbol{X}(t-1), \boldsymbol{X}(t-2), \ldots, \boldsymbol{X}(t-l_x)] \in \boldsymbol{R}^{1\times(l_x\times s)}$,$\overline{\boldsymbol{Y}}(t) = [\boldsymbol{Y}(t-1), \boldsymbol{Y}(t-2), \ldots, \boldsymbol{Y}(t-l_y)] \in \boldsymbol{R}^{1\times(l_y\times r)}$, $\overline{\boldsymbol{A}} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_{l_x}] \in \boldsymbol{R}^{n\times(l_x\times s)}$, and $\overline{\boldsymbol{B}} = \left[\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_{l_y}\right] \in \boldsymbol{R}^{n\times(l_y\times r)}$. $\boldsymbol{D}_x(t) \in \boldsymbol{R}^{1\times s}$ and $\boldsymbol{D}_y(t) \in \boldsymbol{R}^{1\times s}$ denote the disturbance introduced at each time, and it is statistically independent of the past data. The coefficient matrices $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$ can be estimated from the least square method as follows:

$$\begin{cases} \overline{\boldsymbol{A}} = [\overline{\boldsymbol{X}}^{\boldsymbol{T}}(t)\overline{\boldsymbol{X}}(t)]^{-1}\overline{\boldsymbol{X}}^{\boldsymbol{T}}(t)\boldsymbol{X}(t) \\ \overline{\boldsymbol{B}} = [\overline{\boldsymbol{Y}}^{\boldsymbol{T}}(t)\overline{\boldsymbol{Y}}(t)]^{-1}\overline{\boldsymbol{Y}}^{\boldsymbol{T}}(t)\boldsymbol{Y}(t) \end{cases}. \tag{4}$$

Therefore, $\boldsymbol{D}_x(t)$ and $\boldsymbol{D}_y(t)$ can be estimated as follows:

$$\begin{cases} \boldsymbol{D}_x(t) = \boldsymbol{X}(t) - \overline{\boldsymbol{X}}(t)\overline{\boldsymbol{A}} = \boldsymbol{X}(t) - \overline{\boldsymbol{X}}(t)[\overline{\boldsymbol{X}}^{\boldsymbol{T}}(t)\overline{\boldsymbol{X}}(t)]^{-1}\overline{\boldsymbol{X}}^{\boldsymbol{T}}(t)\boldsymbol{X}(t) \\ \boldsymbol{D}_y(t) = \boldsymbol{Y}(t) - \overline{\boldsymbol{Y}}(t)\overline{\boldsymbol{B}} = \boldsymbol{Y}(t) - \overline{\boldsymbol{Y}}(t)[\overline{\boldsymbol{Y}}^{\boldsymbol{T}}(t)\overline{\boldsymbol{Y}}(t)]^{-1}\overline{\boldsymbol{Y}}^{\boldsymbol{T}}(t)\boldsymbol{Y}(t) \end{cases}. \tag{5}$$

Then, the optimal number of time lag will be the one that creates the following indices:

$$\begin{cases} Lag_x = \|\sum_{t=1}^{n} \boldsymbol{D}_x(t)\|^2 = \|\boldsymbol{X} - \overline{\boldsymbol{X}}[\overline{\boldsymbol{X}}^{\boldsymbol{T}}\overline{\boldsymbol{X}}]^{-1}\overline{\boldsymbol{X}}^{\boldsymbol{T}}\boldsymbol{X}\|^2 \\ Lag_y = \|\sum_{t=1}^{n} \boldsymbol{D}_y(t)\|^2 = \|\boldsymbol{Y} - \overline{\boldsymbol{Y}}[\overline{\boldsymbol{Y}}^{\boldsymbol{T}}\overline{\boldsymbol{Y}}]^{-1}\overline{\boldsymbol{Y}}^{\boldsymbol{T}}\boldsymbol{Y}\|^2 \end{cases} \tag{6}$$

the minimum and the indices will not change significantly if we continue increasing the time lag.

As opposed to $\boldsymbol{X}(t)$ and $\boldsymbol{Y}(t)$, $\boldsymbol{D}_x(t)$ and $\boldsymbol{D}_y(t)$ are time-uncorrelated and independent of the initial states of $\boldsymbol{X}(t)$ and $\boldsymbol{Y}(t)$, so they can be adopted to the dynamic process in both steady and unsteady states.

Additionally, we also set up an index to describe 'the value of Lag_x or Lag_y would not change significantly' as shown in Equation (7):

$$RC\% = \frac{|Lag_{i-1} - Lag_i|}{Lag_{i-1}} \times 100\%, \tag{7}$$

where Lag_i represents the value of Lag_x or Lag_y when the lag number is $l_x(l_x > 1)$ or $l_y(l_y > 1)$, and Lag_{i-1} represents the value of Lag_x or Lag_y when the lag number is $l_x - 1$ or $l_y - 1$. If the value of $RC\%$ begins to be less than 5%, we will say that 'the value of Lag_x or Lag_y would not change significantly'.

3.2. DOSA Procedure

Step 1. The "Time Lag Shift" method mentioned in Section 2.2. Calculate the lag number of l_x and l_y in Equation (6). Then, augment $\boldsymbol{X}(t)$ and $\boldsymbol{Y}(t)$ with the previous observations shown in Equation (2). In doing so, we can obtain the augmented matrix $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{Y}}$ with n samples.

Step 2. Traditional OSA mentioned in Section 2.1.

(a) Calculate the Y-related component $\boldsymbol{X}_{OSA} \in \boldsymbol{R}^{n\times[(l_x+1)\times s]}$ and the X-related component $\boldsymbol{Y}_{OSA} \in \boldsymbol{R}^{n\times[(l_y+1)\times s]}$ using Equation (8). $\boldsymbol{X}_{OSA}$ and $\boldsymbol{Y}_{OSA}$ are both called 'the common component' and are shown to be equal in reference [16], as shown below:

$$\begin{cases} \boldsymbol{X}_{OSA} = \widetilde{\boldsymbol{Y}}(\widetilde{\boldsymbol{Y}}^{\boldsymbol{T}}\widetilde{\boldsymbol{Y}})^{-1}\widetilde{\boldsymbol{Y}}^{\boldsymbol{T}}\widetilde{\boldsymbol{X}} \\ \boldsymbol{Y}_{OSA} = \widetilde{\boldsymbol{X}}(\widetilde{\boldsymbol{X}}^{\boldsymbol{T}}\widetilde{\boldsymbol{X}})^{-1}\widetilde{\boldsymbol{X}}^{\boldsymbol{T}}\widetilde{\boldsymbol{Y}} \end{cases}. \tag{8}$$

We tend to focus on process variables related to KPIs in industrial processes. By extracting common components and monitoring them (Step 3), one can know whether there are faults in the variables related to KPIs.

(b) Calculate the non-Y-related component $\boldsymbol{E}_{OSA} \in \boldsymbol{R}^{n\times[(l_x+1)\times s]}$ and the non-X-related component $\boldsymbol{F}_{OSA} \in \boldsymbol{R}^{n\times[(l_y+1)\times s]}$ as

$$\begin{cases} \boldsymbol{E}_{OSA} = \widetilde{\boldsymbol{X}} - \boldsymbol{X}_{OSA} \\ \boldsymbol{F}_{OSA} = \widetilde{\boldsymbol{Y}} - \boldsymbol{Y}_{OSA} \end{cases}, \tag{9}$$

where $\boldsymbol{E}_{OSA}$ and $\boldsymbol{F}_{OSA}$ are both called 'the unique component'. By extracting and monitoring the unique components (Step 3), one can know whether there are faults in the variables unrelated to KPIs.

(c) Extract the PCs in $\boldsymbol{X}_{OSA}$ using the PCA decomposition method because the variables in $\boldsymbol{X}_{OSA}$ might be highly correlated:

$$\begin{cases} \boldsymbol{X}_{OSA} = \boldsymbol{T}_{comx}\boldsymbol{P}_{com}^{\boldsymbol{T}} + \boldsymbol{E}_f \\ \boldsymbol{T}_{comx} = \boldsymbol{X}_{OSA}\boldsymbol{P}_{com} \end{cases}, \tag{10}$$

where $\boldsymbol{T}_{comx} \in \boldsymbol{R}^{n\times k}$ represents the score matrix of the common component; $\boldsymbol{P}_{com} \in \boldsymbol{R}^{[(l_x+1)\times s]\times k}$ is the loading matrix of the common component; $\boldsymbol{E}_f \in \boldsymbol{R}^{n\times[(l_x+1)\times s]}$ is the residual matrix; and k is the number of PCs. In this step, the PCs are selected by using the CPV method, and the threshold value follows the PCA criterion, e.g., 85%.

In theory, the score matrices of the common components $\boldsymbol{X}_{OSA}$ and $\boldsymbol{Y}_{OSA}$ are equal unless there is something wrong with the relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$. We use the sum of squares of the score matrices to monitor whether there are faults in the relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$ (Step 3). Similarly to Equation (10), the score matrix of the common component is $\boldsymbol{T}_{comy} = \boldsymbol{Y}_{OSA}\boldsymbol{P}_{com}$.

Step 3. Monitoring indices calculation.

Taking into account the real-time nature of the information, PCA monitoring is not directly performed for $\boldsymbol{X}_{OSA}$, $\boldsymbol{E}_{OSA}$, and $\boldsymbol{F}_{OSA}$ because these components contain a great amount of information at the past time. The calculation of the indices if as follows:

(a) The first s columns of $\boldsymbol{X}_{OSA}$ are monitored by the PCA approach and can then be used to generate the T_C^2 and SPE_C indices. That is to say, we only monitor the data at the current time.

(b) Similarly, the first s columns of $\boldsymbol{E}_{OSA}$ and the first r columns of $\boldsymbol{F}_{OSA}$ can be monitored by the PCA approach and can then be used to generate the indices T_E^2, T_F^2, SPE_E, and SPE_F.

(c) Furthermore, if there is something wrong with the relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$, there will be significant differences between the score matrices $\boldsymbol{T}_{comx}$ and $\boldsymbol{T}_{comy}$. Therefore, the following index can be used to test the abnormal relationship:

$$SPE_{XY} = (\boldsymbol{T}_{comx} - \boldsymbol{T}_{comy})(\boldsymbol{T}_{comx} - \boldsymbol{T}_{comy})^T. \tag{11}$$

Figure 1 summarizes the procedure presented below.

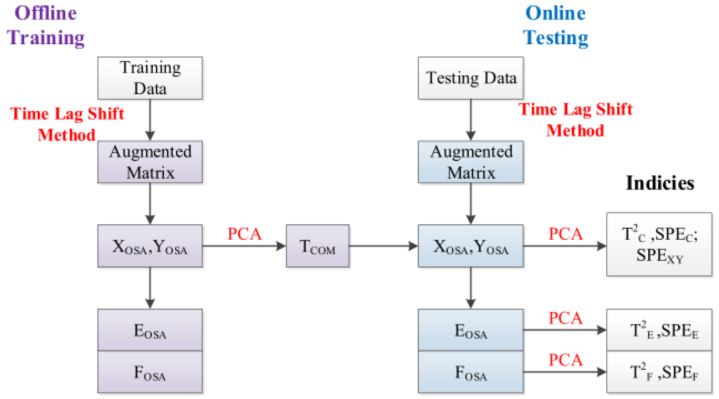

Figure 1. The flow chart of DOSA.

3.3. A Dynamics Model Analyzed with DOSA

3.3.1. Dynamics Model

To analyze the characteristics of the DOSA method and compare its performance with the OSA algorithm, we use a simplistic simulation process in illustrating the monitoring performances of them. Consider a large-scale process in which each single subprocess can be expressed using a time-invariant, state-space model as follows:

$$\begin{cases} \boldsymbol{X}(t) = C[\boldsymbol{X}(t-1), \boldsymbol{X}(t-2), \boldsymbol{X}(t-3)] + \boldsymbol{D}[s_1, s_2] + \xi \\ \boldsymbol{Y}(t) = \boldsymbol{E}[\boldsymbol{Y}(t-1), \boldsymbol{Y}(t-2), \boldsymbol{Y}(t-3)] + \boldsymbol{F}[s_1, s_2] + \zeta \end{cases}, \tag{12}$$

where s_1, s_2, and s_3 are independent Gaussian distributed vectors; ξ and ζ are the noisy components, which are independent of the process measurements; and $\boldsymbol{C}$ and $\boldsymbol{E}$ and $\boldsymbol{D}$

and $\boldsymbol{F}$ are the coefficient matrices of the dynamic and static parts, respectively. Here, we take three algorithms into consideration: OSA; the DOSA that expands the dimension of $\boldsymbol{X}$, which is denoted as DOSA-X; the DOSA that expands the dimension of both $\boldsymbol{X}$ and $\boldsymbol{Y}$, which is denoted as DOSA-XY.

3.3.2. The Optimal Numbers of Time Lag

To determine the number of time lag, the dynamics model with several numbers of lags that are different from the normal data are fitted. Here, l_x and l_y are the numbers of lags in matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. In this work, we set $l_x \in [0.1\ldots,6]$ and $l_y \in [0.1\ldots6]$, and several values of Lag_x and Lag_y are shown in Figures 2 and 3.

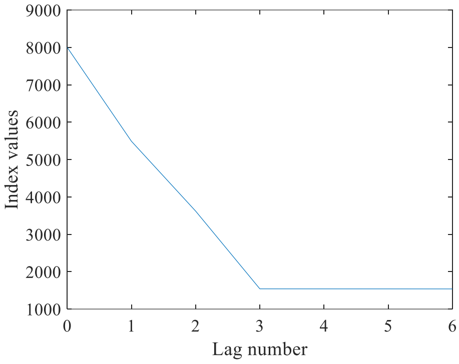

Figure 2. The values of Lag_x under different l_x values.

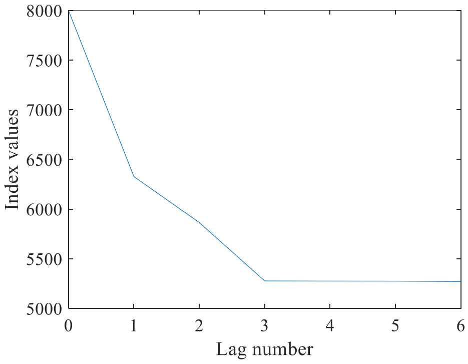

Figure 3. The values of Lag_y under different l_y values.

From the analyses shown in Figures 2 and 3, the values of $Lag_x(l_x = 3)$ would be lowest if l_x was less than or equal to 3, and the values of $Lag_y(l_y = 3)$ tended to be lowest if l_y was less than or equal to 3. At this time, the values of both Lag_x and Lag_y would not decrease significantly if we continued increasing the values of l_x and l_y. Therefore, the optimal lag numbers were $l_x = 3$ and $l_y = 3$, and this can be seen intuitively in the diagram. Furthermore, the several values of Lag_x, Lag_y, and $RC\%$ are listed in Tables 1 and 2.

Table 1. The values of Lag_x under different l_x values.

	$l_x = 0$	$l_x = 1$	$l_x = 2$	$l_x = 3$	$l_x = 4$	$l_x = 5$	$l_x = 6$
Lag_x	8000.4	5489.7	3620.8	**1540.9**	**1540**	**1539.7**	**1538.6**
$RC\%$	/	31.38%	34.04%	57.44%	**0.06%**	**0.19%**	**0.71%**

Table 2. The values of Lag_y under different l_y values.

	$l_y = 0$	$l_y = 1$	$l_y = 2$	$l_y = 3$	$l_y = 4$	$l_y = 5$	$l_y = 6$
Lag_y	7999.1	6327.2	5864.1	**5276.4**	**5275**	**5274.3**	**5270.1**
$RC\%$	/	20.9%	7.32%	10.02%	**0.03%**	**0.01%**	**0.08%**

From the data presented in Tables 1 and 2, the values of $RC\%$ were less than 5% when l_x and l_y gradually increased from 3. This also means that the optimal numbers of lags were l_x = 3 and l_y = 3, which is consistent with the true value.

Here, we take the traditional BIC method, which has a larger penalty than the AIC, as an example to calculate the optimal number of this model. When selecting the best model from a set of alternative models, the model with the lowest BIC should be chosen.

From the data presented in Tables 3 and 4, the optimal numbers of lags were $l_x = 2$ and $l_y = 3$. However, we introduced a third-order lag as Section 3.3.1 mentioned. Therefore, instead of the BIC, the original method of this work was applied to test the algorithm.

Table 3. The values of BIC under different l_x values.

	$l_x = 0$	$l_x = 1$	$l_x = 2$	$l_x = 3$	$l_x = 4$	$l_x = 5$	$l_x = 6$
BIC	−11,427.54	−11,423.19	**−11,453.90**	−11,447.21	−11,442.30	−11,439.81	−11,433.15

Table 4. The values of BIC under different l_y values.

	$l_y = 0$	$l_y = 1$	$l_y = 2$	$l_y = 3$	$l_y = 4$	$l_y = 5$	$l_y = 6$
BIC	−9208.71	−9214.21	−9237.06	**−9237.20**	−9230.39	−9223.74	−9218.98

3.3.3. Testing Results

(a) Fault 1: a step change with an amplitude of 3 in s_1. Certainly, the static parameter s_1 is the unique part of $\boldsymbol{X}$. The detection rates and false alarm rates of three algorithms are shown in Table 5. In Table 5, the detection rate of T_E^2 was extremely high, so we could correctly infer that the fault occurred in the unique part of $\boldsymbol{X}$. In other words, it is possible that there was a fault in the process variables instead of in the measurement of the KPIs. It is more important that the detection rates of the two dynamics monitoring methods were higher than the detection rate of the OSA. Thus, the dynamics problem could be solved by DOSA in this case. Furthermore, the effect of the dimension expansion for both $\boldsymbol{X}$ and $\boldsymbol{Y}$ was better than the dimension expansion for $\boldsymbol{X}$ alone. It can be hypothesized that expanding the dimension of the matrix can improve the sensitivity of the algorithm to the fault.

(b) Fault 2: a step change with an amplitude of 3 in s_3. It is obvious that the static parameter s_3 is the unique part of $\boldsymbol{Y}$. The results are shown in Table 6. As can be seen in Table 6, we had already expanded the dimension of $\boldsymbol{X}$, but the detection rates of all of the indices were extremely low. Then, we found that the index T_F^2 performed better while expanding the dimension of both $\boldsymbol{X}$ and $\boldsymbol{Y}$. This means that the fault occurred in the unique part of $\boldsymbol{Y}$. That is to say, there was a fault in the measurement of the KPIs instead of the process variables. In addition, the detection rate of DOSA-XY was extremely higher than the other two algorithms. Thus, an algorithm for the dimension expansion of data matrices with dynamic processes performs well while also solving the dynamics issue.

Table 5. Fault 1 detection rates and false alarm rates of three algorithms.

Methods	**OSA**						
Indices	T^2_C	SPE_C	T^2_E	SPE_E	T^2_F	SPE_F	SPE_{XY}
Detection rate	1.2	2.4	61.68	15.17	1.8	1	14.97
False alarm rate	1.6	0.6	0.8	0.8	1	0.4	1
Methods	**DOSA-X**						
Indices	T^2_C	SPE_C	T^2_E	SPE_E	T^2_F	SPE_F	SPE_{XY}
Detection rate	1.6	2.2	87.62	55.69	2	1.2	2.4
False alarm rate	1.8	0.6	1.2	2.2	0.8	0.4	0.8
Methods	**DOSA-XY**						
Indices	T^2_C	SPE_C	T^2_E	SPE_E	T^2_F	SPE_F	SPE_{XY}
Detection rate	0.8	1.8	**93.21**	53.29	2.2	1.2	10.58
False alarm rate	0.8	0.4	2.4	1.2	0.4	0.8	0.2

Table 6. Fault 2 detection rates and false alarm rates of three algorithms.

Methods	**OSA**						
Indices	T^2_C	SPE_C	T^2_E	SPE_E	T^2_F	SPE_F	SPE_{XY}
Detection rate	0.6	1	1	0.2	42.91	8.58	32.73
False alarm rate	1	0.8	1.6	1	0.8	1.2	2
Methods	**DOSA-X**						
Indices	T^2_C	SPE_C	T^2_E	SPE_E	T^2_F	SPE_F	SPE_{XY}
Detection rate	0.4	0.8	1	0.8	44.31	5.6	44.71
False alarm rate	1	1.6	1.2	2	1.2	1.2	1
Methods	**DOSA-XY**						
Indices	T^2_C	SPE_C	T^2_E	SPE_E	T^2_F	SPE_F	SPE_{XY}
Detection rate	0.6	0.2	2.4	0.4	**91.82**	9.58	62.48
False alarm rate	2.81	0.6	3.61	1.4	1.6	1.6	1

(c) Fault 3: a step change with an amplitude of 3 in s_2. Certainly, the static parameter s_2 is the common part of both $\boldsymbol{X}$ and $\boldsymbol{Y}$. The results are shown in Table 7. As can be seen in Table 7, we could not judge the location of the fault if we did not expand the dimension of $\boldsymbol{Y}$ because the detection rates of most of the indices were about 50%. Then, the index T^2_C performed better while expanding the dimension of both $\boldsymbol{X}$ and $\boldsymbol{Y}$. This means that the fault occurred in the common part of both $\boldsymbol{X}$ and $\boldsymbol{Y}$. That is to say, there was a fault in both the process variables and in the measurement of the KPIs. In addition, the detection rate of DOSA-XY was extremely higher than that of the other two algorithms. Thus, an algorithm for the dimension expansion of data matrices with dynamic processes performs well while dealing with the dynamics issue.

Table 7. Fault 3 detection rates and false alarm rates of three algorithms.

Methods				**OSA**			
Indices	T_C^2	SPE_C	T_E^2	SPE_E	T_F^2	SPE_F	SPE_{XY}
Detection rate	45.51	29.34	30.94	30.94	45.51	12.38	16.97
False alarm rate	1.4	1	2.61	1.6	1.4	1.2	0.8
Methods				**DOSA-X**			
Indices	T_C^2	SPE_C	T_E^2	SPE_E	T_F^2	SPE_F	SPE_{XY}
Detection rate	45.51	15.17	50.7	52.5	45.51	25.55	1.4
False alarm rate	1.4	1.6	1.2	2	1.4	2.4	1.6
Methods				**DOSA-XY**			
Indices	T_C^2	SPE_C	T_E^2	SPE_E	T_F^2	SPE_F	SPE_{XY}
Detection rate	**90.82**	65.67	7.19	37.72	49.1	52.5	11.98
False alarm rate	3.41	1.4	0.8	2.61	2.2	1.6	1.2

(d) Fault 4: the matrix $\boldsymbol{D}$ changed to $\boldsymbol{D}_f$:

$$\begin{cases} \boldsymbol{D} = \begin{bmatrix} 0.1 & 0.1 & 0.2 & 0 \\ 0.1 & 0 & 0.1 & 0.1 \end{bmatrix} \\ \boldsymbol{D}_f = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0 \\ 0.1 & 0 & 0.1 & 0.1 \end{bmatrix} \end{cases}. \tag{13}$$

Generally, the coefficient matrix $\boldsymbol{D}$ affects the relationship of $\boldsymbol{X}$ and $\boldsymbol{Y}$. The results are shown in Table 8. In Table 8, the index SPE_{XY} that specifically detects the relationship of $\boldsymbol{X}$ and $\boldsymbol{Y}$ performed well. We could infer that there was a high probability of a fault in $\boldsymbol{D}$ or $\boldsymbol{F}$. Then, the detection rate of DOSA-XY was extremely higher than that of the other two algorithms. Thus, an algorithm for the dimension expansion of both $\boldsymbol{X}$ and $\boldsymbol{Y}$ performs well while also solving the dynamics issue.

Table 8. Fault 4 detection rates and false alarm rates of three algorithms.

Methods				**OSA**			
Indices	T_C^2	SPE_C	T_E^2	SPE_E	T_F^2	SPE_F	SPE_{XY}
Detection rate	1.2	1.8	1.4	2	1.2	0.2	64.27
False alarm rate	1.2	0.6	1.2	1.2	1.2	0.8	0.4
Methods				**DOSA-X**			
Indices	T_C^2	SPE_C	T_E^2	SPE_E	T_F^2	SPE_F	SPE_{XY}
Detection rate	1.2	1	0.6	1	1.2	1.6	77.45
False alarm rate	1.2	1.2	1.2	1.8	1.2	1.2	1
Methods				**DOSA-XY**			
Indices	T_C^2	SPE_C	T_E^2	SPE_E	T_F^2	SPE_F	SPE_{XY}
Detection rate	0.4	1.6	1	0.6	1	0.8	**99.6**
False alarm rate	0.4	1.4	2.81	0.4	1.2	1	1.6

3.3.4. The Influence of Sampling Period on DOSA

In sum, it is necessary to expand the dimension of both $\boldsymbol{X}$ and $\boldsymbol{Y}$. In this section, we will take the effect of the sampling rate on the DOSA algorithm into account. The dynamics models and faults in Sections 3.3.1 and 3.3.3 still apply to this section.

Firstly, the section will discuss the effect of doubling the sampling period on the selection of the lag number. We still set $l_x \in [0.1\ldots,6]$ and $l_y \in [0.1\ldots,6]$, followed by several values of Lag_x and Lag_y, and the corresponding changes in rate are listed in Tables 9 and 10.

Table 9. The values of Lag_x for doubling the sampling period.

	$l_x = 0$	$l_x = 1$	$l_x = 2$	$l_x = 3$	$l_x = 4$	$l_x = 5$	$l_x = 6$
Lag_x	501	**202.26**	**194.27**	**186.8**	**183.66**	**179.02**	**176.04**
$RC\%$	/	59.63%	**3.95%**	**3.84%**	**1.68%**	**2.53%**	**1.66%**

Table 10. The values of Lag_y for doubling the sampling period.

	$l_y = 0$	$l_y = 1$	$l_y = 2$	$l_y = 3$	$l_y = 4$	$l_y = 5$	$l_y = 6$
Lag_y	501	**472.91**	**468.78**	**468.64**	**466.8**	**465.47**	**464.08**
$RC\%$	/	5.61%	**0.87%**	**0.03%**	**0.39%**	**0.28%**	**0.30%**

As shown in Tables 9 and 10, the optimal lag numbers were $l_x = 1$ and $l_y = 1$ because the values of $RC\%$ were less than 5% when l_x and l_y gradually increased from 1. That is to say, the optimal lag numbers were affected by the sampling period. Thus, the effect of the sampling period on the detection rates of the DOSA was also a concern.

(a) Fault 1: the fault occurs in the unique part of $\boldsymbol{X}$. The experimental comparison of the primitive and doubled sampling periods is shown in Table 11. As also shown in the table, the detection rate of T_E^2 decreased by about 9%, and the detection rate of SPE_E decreased by about 4%.

Table 11. Comparison of primitive and doubled sampling periods (Fault 1).

Condition	**Primitive sampling period**	
Indices	T_E^2	SPE_E
Detection rate	**93.21**	53.29
False alarm rate	2.4	1.2
Condition	**Doubled sampling period**	
Indices	T_E^2	SPE_E
Detection rate	84.6	49.36
False alarm rate	1.6	1.2

(b) Fault 2: the fault occurs in the unique part of $\boldsymbol{Y}$. The experimental comparison of the primitive and doubled sampling periods is shown in Table 12. As also shown in the table, the detection rate of T_F^2 decreased by about 8%, and the detection rate of SPE_F decreased by about 3%.

(c) Fault 3: the fault occurs in the common part of $\boldsymbol{X}$ and $\boldsymbol{Y}$. The experimental comparison of the primitive and doubled sampling periods is shown in Table 13. As also shown in the table, the detection rate of T_C^2 decreased by about 8%, and the detection rate of SPE_C decreased by about 5%.

Table 12. Comparison of primitive and doubled sampling periods (Fault 2).

Condition	**Primitive sampling period**	
Indices	T_F^2	SPE_F
Detection rate	**91.82**	9.58
False alarm rate	1.6	1.6
Condition	**Doubled sampling period**	
Indices	T_F^2	SPE_F
Detection rate	83.13	6.43
False alarm rate	1.6	1.2

Table 13. Comparison of primitive and doubled sampling periods (Fault 3).

Condition	**Primitive sampling period**	
Indices	T_C^2	SPE_C
Detection rate	**90.82**	65.67
False alarm rate	3.41	1.4
Condition	**Doubled sampling period**	
Indices	T_C^2	SPE_C
Detection rate	82.33	60.84
False alarm rate	2	1.6

(d) Fault 4: the fault occurs in the coefficient matrix $\boldsymbol{D}$, which affects the relationship of $\boldsymbol{X}$ and $\boldsymbol{Y}$. The experimental comparison of the primitive and doubled sampling periods is shown in Table 14. As can be seen in Table 14, there was no significant change in the detection rate of SPE_{XY}.

Table 14. Comparison of primitive and doubled sampling periods (Fault 4).

Condition	**Primitive sampling period**
Indices	SPE_{XY}
Detection rate	**99.6**
False alarm rate	1.6
Condition	**Doubled sampling period**
Indices	SPE_{XY}
Detection rate	**98.39**
False alarm rate	2.4

Based on the above testing results, we can see that the change in sampling period affected the determination of the lag numbers. The detection rates were also slightly affected. That is to say, the DOSA algorithm is sensitive to the change in sampling period because the AR model, which is constructed by the DOSA, will be different with the change in sampling period. We hope to solve this problem as we continue our improvement of this project in the future.

3.4. Conclusion

As shown by the above results, we can conclude the following:

(1) It is necessary to expand the dimension of both $\boldsymbol{X}$ and $\boldsymbol{Y}$.
(2) DOSA could adequately solve the dynamics issue.

(3) DOSA is able to directly analyze the location of the fault. Thus, we can know whether a fault actually occurs in KPI-related process variables, KPI-unrelated process variables, and the measurement of the KPIs.

(4) DOSA is sensitive to the change in sampling period.

4. Comparison Study Based on Tennessee Eastman Process

4.1. Tennessee Eastman Process

In this section, we would like to briefly introduce an industrial benchmark of the Tennessee Eastman (TE) process [24,25]. All the discussed methods will be further applied to demonstrate their efficiencies. The TE process model is a realistic simulation program of a chemical plant, which is widely accepted as a benchmark for control and monitoring studies [26]. The flow diagram of the process is described in [27,28], and the FORTRAN code of the process is available on the Internet. The process has two products from four reactants as shown in Equation (14):

$$\begin{cases} A(g)+C(g)+D(g)\rightarrow G(liq) \\ A(g)+C(g)+E(g)\rightarrow H(liq) \\ A(g)+E(g)\rightarrow F(liq) \\ 3D(g)\rightarrow 2F(liq) \end{cases}, \tag{14}$$

The TE process has 52 variables, including 41 process variables and 11 manipulated variables. Table 15 lists a set of 15 known faults introduced to the TE process. Training and test sets have been collected by running 25 and 48 h simulations, respectively, in which faults have been introduced 1 and 8 h into the simulation, and each variable is sampled every 3 min. Thus, training sets consist of 500 samples, whereas test sets contain 960 samples per set of simulation [29,30].

Table 15. Descriptions of known faults in TE process.

Fault ID	Process Variable	Type	KPI-Related
1	A/C feed ratio, B composition constant	Step	Yes
2	B composition, A/C ration constant		Yes
3	D feed temperature		
4	Reactor cooling water inlet temperature		
5	Condenser cooling water inlet temperature		Yes
6	A feed loss		Yes
7	C header pressure loss-reduced availability		Yes
8	A, B and C feed composition	Random variation	Yes
9	D feed temperature		
10	C feed temperature		Yes
11	Reactor cooling water inlet temperature		
12	Condenser cooling water inlet temperature		Yes
13	Reaction kinetics	Slow drift	Yes
14	Reactor cooling water valve	Sticking	
15	Condenser cooling water valve		

4.2. The Numbers of Time Lag in TE Process

Here, L_x and L_y are the lag numbers in the augmented process variables matrix and the augmented KPI matrix, respectively. In this work, we set $L_x \in [0, 1, \ldots, 6]$ and $L_y \in [0, 1, \ldots, 6]$. Several values of Lag_x and Lag_y and their corresponding changes in rate are listed in Tables 16 and 17.

Table 16. The values of Lag_x under different L_x values.

	$L_x = 0$	$L_x = 1$	$L_x = 2$	$L_x = 3$	$L_x = 4$	$L_x = 5$	$L_x = 6$
Lag_x	159	62.95	28.32	**3.6**	13,104.51	66,817.24	34,678.86

Table 17. The values of Lag_y under different L_y values.

	$L_y = 0$	$L_y = 1$	$L_y = 2$	$L_y = 3$	$L_y = 4$	$L_y = 5$	$L_y = 6$
Lag_y	159	112.86	100.61	**85.87**	**83.91**	**79.84**	**79**
RC%	/	29.02%	10.85%	14.65%	**2.28%**	**4.85%**	**1.05%**

From the data presented in Tables 16 and 17, the values of $Lag_x(L_x = 3)$ tended to be the lowest, and the values of $Lag_y(L_y = 3)$ tended to be the lowest if L_y was less than or equal to 3. At this time, the values of the rate of change were less than 5% when L_y gradually increased from 3. That is to say, the values of Lag_y would not decrease significantly if we continued increasing L_y. Therefore, the optimal numbers of lags were $L_x = 3$ and $L_y = 3$.

4.3. Simulation Study

We tend to focus on the ability to detect KPI-related faults in the TE process. Table 18 lists a set of nine KPI-related faults introduced to the TE process. It shows the detection and false alarm rates for four algorithms: OSA, DOSA, Dynamics CCA (DCCA), and Dynamics PLS (DPLS).

Table 18. Testing results of KPI-related faults for the TE process.

	DPLS	**DCCA**		**OSA**		**DOSA**	
	$\boldsymbol{T^2}$	$\boldsymbol{SPE_1}$	$\boldsymbol{SPE_2}$	$\boldsymbol{T_C^2}$	$\boldsymbol{SPE_C}$	$\boldsymbol{T_C^2}$	$\boldsymbol{SPE_C}$
False alarm rate	0	1.3	1.3	0	0	0	0.63
Fault 1	42.625	73.7	**91.4**	61.75	88.25	**99.375**	**97.375**
Fault 2	**98.75**	86	89	15.375	53.75	**97.125**	**96.375**
Fault 5	20.125	**98.9**	**99.9**	16.875	11.25	22.375	15.125
Fault 6	**96.5**	**100**	**100**	**99.125**	**100**	**100**	**100**
Fault 7	38	17.5	34.5	21.5	89	63.75	29.125
Fault 8	68	43.3	53.1	67	51.625	**92.875**	74.75
Fault 10	5.375	21.9	37.2	60.875	13.125	66.875	70.25
Fault 12	31	66.2	85.2	69.125	51.125	**94.625**	77.375
Fault 13	66.125	78.6	85.2	80	70.625	**90.75**	76.625

Considering the data presented in Table 18, DOSA shows better performance compared to the other algorithms for KPI-related faults. Meanwhile, the DOSA algorithm showed a great advantage in Faults 1–2, 8, and 12–13 over the OSA algorithm. From this analysis, it can be concluded that the DOSA algorithm performs better than the OSA algorithm on dynamic problems. Figure 4 shows the simulation diagram of OSA and DOSA monitoring in these faults. The blue line represents the value of the statistic, and the red line represents the value of the control limit. When the blue line is higher than the red line, a fault has occurred. It is obvious that the DOSA algorithm is more sensitive to these faults.

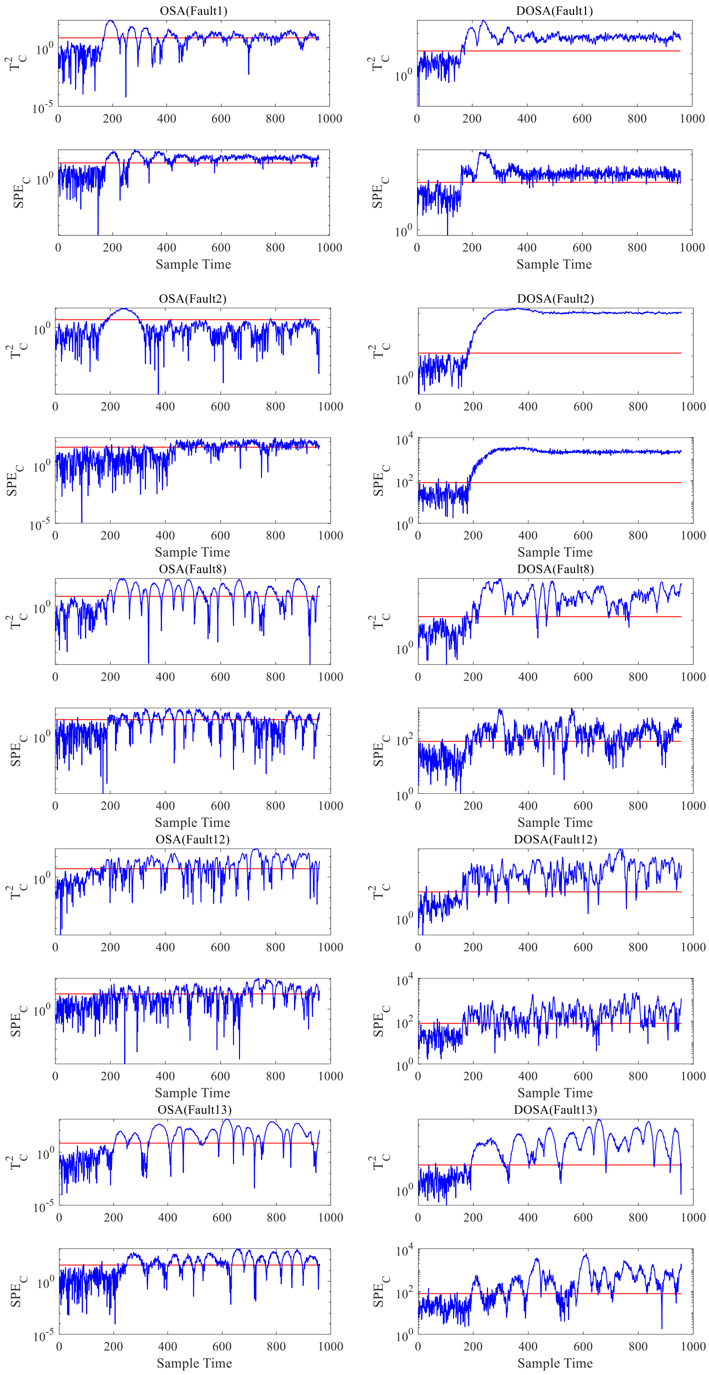

Figure 4. The simulation comparison of OSA and DOSA monitoring in Faults 1–2, 8, and 12–13.

5. Conclusions

In this paper, we have presented an improved algorithm of OSA for conducting large-scale process monitoring, called the DOSA algorithm, and compared its performance against DPLS and DCCA, which are KPI-related algorithms that are also used to solve dynamic problems.

Considering the testing results of the dynamics model, this article proved that it is necessary to expand the dimension of both the process variables matrix and the KPI matrix while using the DOSA algorithm. Furthermore, the DOSA algorithm is able to adequately solve the dynamics issue; Thus, we can know whether a fault actually occurs in the KPI-related or KPI-unrelated process variables or in the measurement of the KPIs.

The comparative study was conducted using the Tennessee Eastman benchmark process, and we can conclude that the DOSA algorithm achieves better detection rates of faults from the analysis of the results obtained. However, the DOSA algorithm is sensitive to the change in sampling period. We intend to solve this problem as we continue the improvement of this project in the future.

Author Contributions: Conceptualization, W.H.; methodology, W.H. and Z.L.; validation, S.L.; formal analysis, W.H.; resources, Y.W. and S.L.; writing—original draft preparation, W.H.; writing—review and editing, Z.L. and W.H.; visualization, W.H.; supervision, S.L., X.J., and S.D.; project administration, Z.L.; funding acquisition, S.L. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Guangdong Province, China (NO. 2022A1515011040), the Natural Science Foundation of Shenzhen, China (NO. 20220813001358001) and the Young Talents program offered by the Department of Education of Guangdong Province, China (2021KQNCX210).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, J.; Jiang, M.; Liu, Z. Fault Detection and Diagnosis in Industrial Processes with Variational Autoencoder: A Comprehensive Study. *Sensors* **2022**, *22*, 227. [CrossRef]
2. Zhao, F.; Rekik, I.; Lee, S.W.; Liu, J.; Zhang, J.; Shen, D. Two-Phase Incremental Kernel PCA for Learning Massive or Online Datasets. *Complexity* **2019**, *2019*, 5937274. [CrossRef]
3. Zhang, S.; Zhao, C. Hybrid Independent Component Analysis (H-ICA) with Simultaneous Analysis of High-Order and Second-Order Statistics for Industrial Process Monitoring. *Chemom. Intell. Lab. Syst.* **2019**, *185*, 47–58. [CrossRef]
4. Qin, Y.; Lou, Z.; Wang, Y.; Lu, S.; Sun, P. An Analytical Partial Least Squares Method for Process Monitoring. *Control. Eng. Pract.* **2022**, *124*, 105182. [CrossRef]
5. Yin, S.; Zhu, X.; Kaynak, O. Improved PLS Focused on Key-Performance-Indicator-Related Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2015**, *62*, 1651–1658. [CrossRef]
6. Wang, H.; Gu, J.; Wang, S.; Saporta, G. Spatial Partial Least Squares Autoregression: Algorithm and Applications. *Chemom. Intell. Lab. Syst.* **2019**, *184*, 123–131. [CrossRef]
7. Tao, Y.; Shi, H.; Song, B.; Tan, S. Parallel Quality-Related Dynamic Principal Component Regression Method for Chemical Process Monitoring. *J. Process Control* **2019**, *73*, 33–45. [CrossRef]
8. Sim, S.F.; Jeffrey Kimura, A.L. Partial Least Squares (PLS) Integrated Fourier Transform Infrared (FTIR) Approach for Prediction of Moisture in Transformer Oil and Lubricating Oil. *J. Spectrosc.* **2019**, *2019*, e5916506. [CrossRef]
9. Kanatsoulis, C.I.; Fu, X.; Sidiropoulos, N.D.; Hong, M. Structured SUMCOR Multiview Canonical Correlation Analysis for Large-Scale Data. *IEEE Trans. Signal Process.* **2019**, *67*, 306–319. [CrossRef]
10. Cai, J.; Dan, W.; Zhang, X. $\ell 0$-Based Sparse Canonical Correlation Analysis with Application to Cross-Language Document Retrieval. *Neurocomputing* **2019**, *329*, 32–45. [CrossRef]
11. Su, C.H.; Cheng, T.W. A Sustainability Innovation Experiential Learning Model for Virtual Reality Chemistry Laboratory: An Empirical Study with PLS-SEM and IPMA. *Sustainability* **2019**, *11*, 1027. [CrossRef]
12. Alvarez, A.; Boente, G.; Kudraszow, N. Robust Sieve Estimators for Functional Canonical Correlation Analysis. *J. Multivar. Anal.* **2019**, *170*, 46–62. [CrossRef]
13. de Cheveigné, A.; Di Liberto, G.M.; Arzounian, D.; Wong, D.D.E.; Hjortkjær, J.; Fuglsang, S.; Parra, L.C. Multiway Canonical Correlation Analysis of Brain Data. *NeuroImage* **2019**, *186*, 728–740. [CrossRef]
14. Tong, C.; Lan, T.; Yu, H.; Peng, X. Distributed Partial Least Squares Based Residual Generation for Statistical Process Monitoring. *J. Process Control* **2019**, *75*, 77–85. [CrossRef]
15. Si, Y.; Wang, Y.; Zhou, D. Key-Performance-Indicator-Related Process Monitoring Based on Improved Kernel Partial Least Squares. *IEEE Trans. Ind. Electron.* **2021**, *68*, 2626–2636. [CrossRef]
16. Lou, Z.; Wang, Y.; Si, Y.; Lu, S. A Novel Multivariate Statistical Process Monitoring Algorithm: Orthonormal Subspace Analysis. *Automatica* **2022**, *138*, 110148. [CrossRef]
17. Song, Y.; Liu, J.; Chu, N.; Wu, P.; Wu, D. A Novel Demodulation Method for Rotating Machinery Based on Time-Frequency Analysis and Principal Component Analysis. *J. Sound Vib.* **2019**, *442*, 645–656. [CrossRef]
18. Zhang, C.; Guo, Q.; Li, Y. Fault Detection Method Based on Principal Component Difference Associated with DPCA. *J. Chemom.* **2019**, *33*, e3082. [CrossRef]

19. Dong, Y.; Qin, S.J. A Novel Dynamic PCA Algorithm for Dynamic Data Modeling and Process Monitoring. *J. Process Control* **2018**, *67*, 1–11. [CrossRef]
20. Oyama, D.; Kawai, J.; Kawabata, M.; Adachi, Y. Reduction of Magnetic Noise Originating from a Cryocooler of a Magnetoencephalography System Using Mobile Reference Sensors. *IEEE Trans. Appl. Supercond.* **2022**, *32*, 1–5. [CrossRef]
21. Lou, Z.; Shen, D.; Wang, Y. Two-step Principal Component Analysis for Dynamic Processes Monitoring. *Can. J. Chem. Eng.* **2018**, *96*, 160–170. [CrossRef]
22. Sakamoto, W. Bias-reduced Marginal Akaike Information Criteria Based on a Monte Carlo Method for Linear Mixed-effects Models. *Scand. J. Stat.* **2019**, *46*, 87–115. [CrossRef]
23. Gu, J.; Fu, F.; Zhou, Q. Penalized Estimation of Directed Acyclic Graphs from Discrete Data. *Stat. Comput.* **2019**, *29*, 161–176. [CrossRef]
24. Wan, J.; Li, S. Modeling and Application of Industrial Process Fault Detection Based on Pruning Vine Copula. *Chemom. Intell. Lab. Syst.* **2019**, *184*, 1–13. [CrossRef]
25. Huang, J.; Ersoy, O.K.; Yan, X. Fault Detection in Dynamic Plant-Wide Process by Multi-Block Slow Feature Analysis and Support Vector Data Description. *ISA Trans.* **2019**, *85*, 119–128. [CrossRef]
26. Plakias, S.; Boutalis, Y.S. Exploiting the Generative Adversarial Framework for One-Class Multi-Dimensional Fault Detection. *Neurocomputing* **2019**, *332*, 396–405. [CrossRef]
27. Zhao, H.; Lai, Z. Neighborhood Preserving Neural Network for Fault Detection. *Neural Netw.* **2019**, *109*, 6–18. [CrossRef]
28. Suresh, R.; Sivaram, A.; Venkatasubramanian, V. A Hierarchical Approach for Causal Modeling of Process Systems. *Comput. Chem. Eng.* **2019**, *123*, 170–183. [CrossRef]
29. Amin, M.T.; Khan, F.; Imtiaz, S. Fault Detection and Pathway Analysis Using a Dynamic Bayesian Network. *Chem. Eng. Sci.* **2019**, *195*, 777–790. [CrossRef]
30. Cui, P.; Zhan, C.; Yang, Y. Improved Nonlinear Process Monitoring Based on Ensemble KPCA with Local Structure Analysis. *Chem. Eng. Res. Des.* **2019**, *142*, 355–368. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.