Deep Canonical Correlation Analysis Using Sparsity-Constrained Optimization for Nonlinear Process Monitoring

Xianchao Xiu, *Member, IEEE*, Zhonghua Miao, Ying Yang, *Senior Member, IEEE*, and Wanquan Liu, *Senior Member, IEEE*

***Abstract*—This article proposes an efficient nonlinear process monitoring method (DCCA-SCO) by integrating canonical correlation analysis (CCA), deep autoencoder neural networks (DAENNs), and sparsity-constrained optimization (SCO). Specifically, DAENNs are first used to learn a nonlinear function automatically, which characterizes intrinsic features of the original process data. Then, the CCA is performed in that low-dimensional representation space to extract the most correlated variables. In addition, the SCO is imposed to reduce the redundancy of the hidden representation. Unlike other deep CCA methods, the DCCA-SCO provides a new nonlinear method that is able to learn a nonlinear mapping with a sparse prior. The validity of the proposed DCCA-SCO is extensively demonstrated on the benchmark Tennessee Eastman (TE) process and the diesel generator process. In particular, compared with the classical CCA, the fault detection rate is increased by 8.00% for the fault IDV(11) in the TE process.**

***Index Terms*—Canonical correlation analysis (CCA), deep autoencoder neural networks (DAENNs), process monitoring (PM), sparsity-constrained optimization (SCO).**

I. Introduction

With the development of intelligent manufacturing, artificial intelligence and machine intelligence play a very important role in modern industry. Machine vision, as the application of machine intelligence in electronic engineering, is often used for automatic detection, process monitoring (PM), and robot guidance. Without exaggeration, it is possible to claim that PM is crucial in satisfying the high requirements for product quality, efficiency maintenance, production costs, safety levels, and environmental protection [1]. Compared with model-based PM, data-driven PM becomes more dominated and efficient due to the development of big-data computing and modern sensors (see [2], [3], and references therein). The most popular data-driven PM methods involve principal component analysis (PCA) [4], [5], independent component analysis [6], partial least squares (PLS) [7], and canonical correlation analysis (CCA) [8]. From the perspective of representation learning, CCA is different from other methods because it integrates inputs and outputs of the process data, which can provide an effective way to explore the relationship between process variables and to allow for better monitoring. Therefore, CCA-based PM has attracted increasing attention in both research community and industrial application [9]–[12].

CCA can be traced back to Hotelling [13], who proposed it for finding a pair of canonical matrices such that the projected variables have maximum correlations. The interested readers can refer to the work in [14] for a comprehensive review and its various modifications, extensions, and generalizations. Even though CCA has been widely used in computer vision and information sciences, the application to PM was not investigated sufficiently. The earlier work only used CCA to perform system identification, filtering, and adaptive control [15], [16]. Recently, a residual-based CCA framework was developed, which can achieve higher monitoring performance than PCA and PLS [8]. Since then, CCA-based PM has been well recognized, including improved CCA [9], distributed CCA [10], multimode CCA [11], and quality-based CCA [12], to name just a few.

However, for nonlinear processes, the aforementioned CCA-based PM methods perform poorly owing to its assumption that the relationship among latent process variables is linear. To alleviate this issue, a kernel variant of CCA, called KCCA, was proposed to extract the canonical correlations for nonlinear processes [17]. The main idea is to use a kernel function to project the original nonlinear process data onto a higher dimensional space, in which the projected data can be separated by linear classifiers [18]. Although KCCA can capture the nonlinear input–output relationship, it has an obvious drawback, which is that the kernel function and parameters need to be carefully

Manuscript received June 27, 2021; revised September 29, 2021; accepted October 14, 2021. Date of publication October 21, 2021; date of current version July 11, 2022. This work was supported by the National Natural Science Foundation of China under Grant 12001019, Grant 61633001, and Grant 62173003. Paper no. TII-21-2666. *(Corresponding author: Ying Yang.)*

Xianchao Xiu and Zhonghua Miao are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China (e-mail: xcxiu@shu.edu.cn; zhhmiao@shu.edu.cn).

Ying Yang is with the State Key Laboratory for Turbulence and Complex Systems, Department of Mechanics and Engineering Science, College of Engineering, Peking University, Beijing 100871, China (e-mail: yy@pku.edu.cn).

Wanquan Liu is with the School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou 510275, China (e-mail: liuwq63@mail.sysu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3121770.

Digital Object Identifier 10.1109/TII.2021.3121770

selected. In addition, calculating kernel matrices always takes a long time, which poses a barrier to these kernel-based PM methods. Due to the rapid growth of neural networks, nonlinear functions can be learned automatically from the given process data. It is also admitted that compared with artificial neural networks, deep neural networks (DNNs) have the superiority in learning representations from the industrial processes. In this sense, a DNN-based CCA (DCCA) framework was constructed, which can obtain higher correlations than those learned by the existing CCA and KCCA [19], [20]. Prior work has shown the benefit of deep autoencoder neural networks (DAENNs), such as deep coupling DAENNs [21] and deep nonnegative matrix factorization [22]. Therefore, a natural question is whether CCA and DAENNs can be combined and then applied to PM.

As pointed out by Wu *et al.* [23], modern industrial processes appear to have redundancy because not all the process variables are identically informative for representing the canonical variables. This means that the aforementioned CCA, KCCA, and DCCA may derive a solution in lack of physical representation. In order to obtain a consistent representation, some sparse regularization terms or constraints can be added [24]. It is worthwhile emphasizing that instead of using convex relaxations, the ℓ_0-norm (number of nonzero elements) sparsity-constrained optimization (SCO) provides a better choice [25]. Although it is NP-hard in general, there still exist some efficient solvers [26]. The reason why this original sparse description is used lies in 1) easier to control variables and 2) faster optimization algorithms available. However, the integration of the ℓ_0-norm and CCA has not been investigated in the literature. Meanwhile, for PM applications, each row is associated with an original process variable, and a zero row indicates that this variable may not be as important as other variables, or even fault-free variables [4]. Naturally, it will be much better if the joint sparse structures are considered in the optimization [27]. Therefore, it is necessary to derive a CCA variant that not only can extract meaningful variables, but also learn deep correlations.

In this article, an efficient nonlinear data-driven PM method based on CCA is proposed, which consists of DAENNs and SCO, referred to as DCCA-SCO. Fig. 1 presents a general idea of the proposed DCCA-SCO, whose detailed explanation is given in Section III. In comparison with CCA-based PM methods, DCCA-SCO constructs a different deep learning framework for nonlinear industrial processes, which can automatically learn appropriate nonlinear representations from process data. Moreover, the SCO is introduced to reduce the effect of outliers and remove some unimportant variables from the projection space. To the best of the authors' knowledge, it is the first time to integrate the SCO into a DCCA framework in order to improve the monitoring performance for nonlinear processes. The contributions of this proposed DCCA-SCO are as follows.

1) A new data-driven PM method is proposed by considering the advantages of DAENNs.
2) The SCO technique is incorporated to enhance the representation and reduce the redundant information.
3) An efficient algorithm is developed and its effectiveness is verified on the benchmark simulated Tennessee Eastman (TE) process and a practical diesel generator (DG) process.

Fig. 1. Framework of DCCA-SCO.

II. Background

A. Canonical Correlation Analysis

Assume that the process data of N samples are given by

$$\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_N] \in \mathbb{R}^{p\times N}$$
$$\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2\ \cdots\ \mathbf{y}_N] \in \mathbb{R}^{q\times N} \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^p$ and $\mathbf{y}_i \in \mathbb{R}^q$ denote the ith samples of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Mathematically, CCA can be presented by the following optimization model:

$$\min_{\mathbf{A},\mathbf{B}} \ -\mathrm{Tr}(\mathbf{A}^T\mathbf{X}\mathbf{Y}^T\mathbf{B})$$
$$\text{s.t.}\ \ \mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} = \mathbf{I},\ \mathbf{B}^T\mathbf{Y}\mathbf{Y}^T\mathbf{B} = \mathbf{I} \quad (2)$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_r] \in \mathbb{R}^{p\times r}$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r] \in \mathbb{R}^{q\times r}$ are the CCA directions, and $\mathbf{I}$ is the identity matrix with rank-r. Note that r is the dimension reduced by CCA, which is usually less than p and q. The core of CCA is to consider the relationship between $\mathbf{X}$ and $\mathbf{Y}$ so as to achieve better representation by exploiting different but complementary information.

A popular nonlinear variant of CCA is called KCCA, which is frequently investigated because of its simple formulation and easy implementation. KCCA projects the process data onto a high-dimensional space through a nonlinear mapping function and extracts the canonical correlations on that shared high-dimensional space, in which the nonlinearity can be reduced. KCCA is

$$\min_{\mathbf{A},\mathbf{B}} \ -\mathrm{Tr}(\mathbf{A}^T\mathbf{K}_\mathrm{X}\mathbf{K}_\mathrm{Y}^T\mathbf{B})$$
$$\text{s.t.}\ \ \mathbf{A}^T\mathbf{K}_\mathrm{X}\mathbf{K}_\mathrm{X}^T\mathbf{A} = \mathbf{I},\ \mathbf{B}^T\mathbf{K}_\mathrm{Y}\mathbf{K}_\mathrm{Y}^T\mathbf{B} = \mathbf{I} \quad (3)$$

where $\mathbf{K}_\mathrm{X}$ and $\mathbf{K}_\mathrm{Y}$ are the kernel matrices of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Since it is based on a kernel function, KCCA can capture the nonlinear relationship among process variables. However,

KCCA involves the computation of eigenvalues, which is challenging in both storing the kernel matrices and calculating matrix decomposition in high-dimensional settings. In addition, it is troublesome to choose an appropriate kernel.

B. Deep Autoencoder Neural Networks

As a DNN-based algorithm, DAENNs can learn a nonlinear function automatically to make the original data and reconstructed data as close as possible, which can be used to extract the most important features of the original data. Now, the DAENNs have been extensively used in image processing, language translation, and industrial informatics. In general, the DAENNs consist of two parts: the encoder and the decoder. The encoder projects the original data onto a low-dimensional representation space by a neural network. However, KCCA projects the original data onto a high-dimensional space by a kernel function, which has many differences. The decoder projects that low-dimensional representation data back onto the reconstructed space, which can be seen as the inverse procedure of the aforementioned encoder procedure. Thus, DAENNs can approximate nonlinear functions to obtain better data representations. See Goodfellow *et al.* [28] for further detailed information.

The reconstruction error can be obtained by summing the differences between the original input data $\mathbf{x}$ and the reconstructed data $\mathbf{x}'$. To improve the robustness of DAENNs, the following loss function is often considered:

$$\text{LOSS} = \frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2 + \lambda\Omega \tag{4}$$

where Ω is a sparse regularization term for variable selection, and λ is a parameter to tune the sparsity. As mentioned by Ng [29], when the output value is close to 0, neurons are inactive and need to be discarded.

III. Methodology

This section presents a new deep CCA framework that incorporates CCA, DAENNs, and the SCO. Then, a fault detection scheme based on the proposed DCCA-SCO is established step-by-step.

A. DCCA-SCO

First, recall the framework of the proposed DCCA-SCO, as shown in Fig. 1. There exist two different input variables, i.e., $\mathbf{X}$ and $\mathbf{Y}$, and the final goal is achieving data fusion to perform PM. The procedure can be roughly divided into three modules: the encoder module, the CCA module, and the decoder module, which are discussed in detail as follows.

1) Encoder Module: Unlike linear CCA, DCCA-SCO first applies a nonlinear transformation to project the given process data onto a low-dimensional representation space. The nonlinearity, denoted as the encoder $\text{En}(\cdot)$, can be estimated by DNNs rather than kernel functions for KCCA. For given data $\mathbf{x} \in \mathbb{R}^p$, the input data $\mathbf{x}$ can be transformed to

$$f(\mathbf{x}) = \text{En}(\mathbf{x}) \tag{5}$$

which is given by

$$\begin{cases} \mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\ \mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2) \\ \cdots\cdots \\ \mathbf{h}_n = \sigma(\mathbf{W}_n\mathbf{h}_{n-1} + \mathbf{b}_n) \end{cases} \tag{6}$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weighted matrix and biased vector of the encoding procedure, and σ is an activation function, including a sigmoid function, a tangent function, and a rectified linear unit (ReLU) function. Similarly, the input data $\mathbf{y}$ can be transformed to

$$g(\mathbf{y}) = \text{En}(\mathbf{y}). \tag{7}$$

After the encoder network is created, the nonlinear functions (f, g) can automatically learn the latent nonlinear structures from the input process data. While, for the classical kernel-based methods, the kernel functions and kernel parameters should be carefully selected, which often takes a lot of time.

2) CCA Module: When the process data $\mathbf{X}$ and $\mathbf{Y}$ have been projected to $f(\mathbf{X})$ and $g(\mathbf{Y})$, linear CCA is then used to calculate the canonical matrices on that low-dimensional representation space. However, some process variables are redundant or not informative to the fault, which should be removed from the transformations.

To improve the representation and generalization, the SCO can be applied to encourage the sparsity of canonical matrices. Considering the fact that each row is related to one specific process variable, the SCO with joint sparsity can improve the performance of PM because the fault-free variables will be removed from the projection space [30]. Therefore, the proposed CCA can be defined as

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{B}} \ -\text{Tr}(\mathbf{A}^T f(\mathbf{X}) g(\mathbf{Y})^T \mathbf{B}) \\ &\text{s.t.} \ \ \mathbf{A}^T f(\mathbf{X}) f(\mathbf{X})^T \mathbf{A} = \mathbf{I}, \ \mathbf{B}^T g(\mathbf{Y}) g(\mathbf{Y})^T \mathbf{B} = \mathbf{I} \\ &\qquad \|\mathbf{A}\|_{2,0} \le s_1, \|\mathbf{B}\|_{2,0} \le s_2 \end{aligned} \tag{8}$$

where $\|\cdot\|_{2,0}$ denotes the $\ell_{2,0}$-norm, which counts the number of nonzero rows, and s_1 and s_2 are the sparsity level to determine how many process variables should be preserved. In fact, the proposed model (8) is a generalized version of the aforementioned CCA model (2) and KCCA model (3), as well as embeds the SCO with joint sparsity.

For presentation clarity, Fig. 2 shows the advantages of the proposed method, where a white block indicates a zero component and a gray block indicates a nonzero component. It is found that the classical sparse CCA can obtain sparsity on each block, whereas the proposed CCA can obtain sparsity on rows and keep the most important s variables by applying the $\ell_{2,0}$-norm constraint. For example, if two important variables need to be explored, just set $s = 2$ [see Fig. 2(b)].

For ease of expression, denote

$$\text{loss1} = -\text{Tr}(\mathbf{A}^T f(\mathbf{X}) g(\mathbf{Y})^T \mathbf{B}). \tag{9}$$

3) Decoder Module: After data fusion, the decoder module reprojects the representation $f(\mathbf{x})$ and $g(\mathbf{y})$ back onto the original space, denoted as $\mathbf{x}'$ and $\mathbf{y}'$. For the representation $f(\mathbf{x})$, the

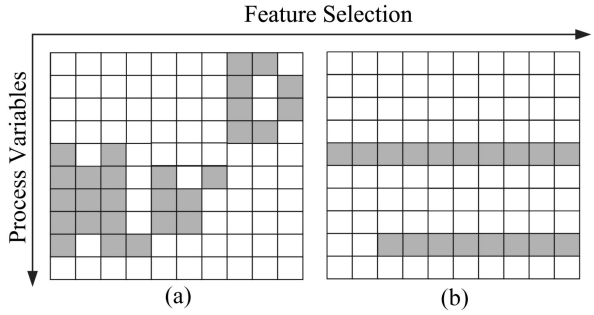


Fig. 2. Selected variables by (a) the classical sparse CCA and (b) the proposed CCA.

Algorithm 1: An Iterative Hard Thresholding Method for (15).

Input: Given data $f(\mathbf{X})$ and $\mathbf{C}^k$, sparsity level $s_1 > 0$, and step-size $\eta > 0$

Output: $\mathbf{A}$

While not converged **do**

1: Compute

$$\mathbf{A}^{k+1} = \mathbf{A}^k - \eta f(\mathbf{X})(f(\mathbf{X})^T\mathbf{A}^k - \mathbf{C}^k)$$

2: Truncate $\mathbf{A}^{k+1}$ with s_1 top (in row-level) entries preserved

End while

decoder

$$\mathbf{x}' = \mathrm{De}(f(\mathbf{x})) \tag{10}$$

can be obtained through

$$\begin{cases} \mathbf{h}_1' = \sigma'(\mathbf{W}_1' f(\mathbf{x}) + \mathbf{b}_1') \\ \mathbf{h}_2' = \sigma'(\mathbf{W}_2' \mathbf{h}_1' + \mathbf{b}_2') \\ \cdots\cdots \\ \mathbf{x}' = \sigma'(\mathbf{W}_n' \mathbf{h}_{n-1}' + \mathbf{b}_n') \end{cases} \tag{11}$$

where $\mathbf{W}_i'$, $\mathbf{b}_i'$, and σ' are the weighted matrix, biased vector, and activation function of the decoding procedure, respectively. For the representation $g(\mathbf{y})$, a similar decoder strategy can be applied. After the decoder network is constructed, a pair of nonlinear functions (p, q) can automatically reconstruct the input process data.

To measure the reconstruction performance, the loss functions can be given by

$$\begin{aligned} \text{loss2} &= \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{x}_i - p(f(\mathbf{x}_i))\|^2 \\ \text{loss3} &= \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - q(g(\mathbf{y}_i))\|^2. \end{aligned} \tag{12}$$

Finally, the whole optimization model of the proposed DCCA-SCO framework can be expressed as

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{B}} \ \text{LOSS} \\ &\text{s.t.} \ \ \mathbf{A}^T f(\mathbf{X}) f(\mathbf{X})^T \mathbf{A} = \mathbf{I}, \ \mathbf{B}^T g(\mathbf{Y}) g(\mathbf{Y})^T \mathbf{B} = \mathbf{I} \\ &\qquad \|\mathbf{A}\|_{2,0} \le s_1, \|\mathbf{B}\|_{2,0} \le s_2 \end{aligned} \tag{13}$$

where $\text{LOSS} = \text{loss1} + \beta_1 \text{loss2} + \beta_2 \text{loss3}$ with $\beta_1, \beta_2 > 0$ being the tradeoff parameters.

It is found that the problem (13) is not easy to solve because all the constraints are nonconvex and the objective function involves two variables. Inspired by Boyd *et al.* [31], it can be computed by solving one variable with the other one fixed. First, two auxiliary variables $\mathbf{C}$ and $\mathbf{D}$ are introduced to make the constraints separable, i.e.,

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} \ \text{LOSS} \\ &\text{s.t.} \ \ \|\mathbf{A}\|_{2,0} \le s_1, \|\mathbf{B}\|_{2,0} \le s_2 \\ &\qquad \mathbf{C}^T\mathbf{C} = \mathbf{I}, \ \mathbf{D}^T\mathbf{D} = \mathbf{I} \\ &\qquad f(\mathbf{X})^T\mathbf{A} = \mathbf{C}, \ g(\mathbf{Y})^T\mathbf{B} = \mathbf{D} \end{aligned} \tag{14}$$

which can be equivalently rewritten as the following regularized version:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} \ \text{LOSS} \\ &\qquad + \frac{\mu_1}{2}\|f(\mathbf{X})^T\mathbf{A} - \mathbf{C}\|_F^2 + \frac{\mu_2}{2}\|g(\mathbf{Y})^T\mathbf{B} - \mathbf{D}\|_F^2 \\ &\text{s.t.} \ \ \|\mathbf{A}\|_{2,0} \le s_1, \ \|\mathbf{B}\|_{2,0} \le s_2 \\ &\qquad \mathbf{C}^T\mathbf{C} = \mathbf{I}, \ \mathbf{D}^T\mathbf{D} = \mathbf{I} \end{aligned} \tag{15}$$

where $\mu_1, \mu_2 > 0$ are two penalty parameters to control the regularization terms. It should also be noted that loss1 has become $-\mathrm{Tr}(\mathbf{C}^T\mathbf{D})$ in this case.

Next, the minimizations for $\mathbf{A}$ and $\mathbf{C}$ will be presented, and the algorithms work similarly for $\mathbf{B}$ and $\mathbf{D}$.

1) To optimize $\mathbf{A}$, the minimization can be simplified to

$$\begin{aligned} &\min_{\mathbf{A}} \ \frac{1}{2}\|f(\mathbf{X})^T\mathbf{A} - \mathbf{C}^k\|_F^2 \\ &\text{s.t.} \ \ \|\mathbf{A}\|_{2,0} \le s_1. \end{aligned} \tag{16}$$

Due to the presence of $\|\mathbf{A}\|_{2,0} \le s_1$, problem (16) is NP-hard [32]. However, an efficient iterative hard thresholding method can be used to estimate $\mathbf{A}$. It first computes the gradient descent with step size $\eta > 0$, and then applies a simple rowwise truncation operation to obtain its approximate solution, which is given in Algorithm 1.

2) To optimize $\mathbf{C}$, after some trivial manipulations, the minimization can be transformed as

$$\begin{aligned} &\min_{\mathbf{C}} \ \frac{1}{2}\|\mathbf{C} - (f(\mathbf{X})^T\mathbf{A} + \mathbf{D}^k/\mu_1)\|_F^2 \\ &\text{s.t.} \ \ \mathbf{C}^T\mathbf{C} = \mathbf{I}. \end{aligned} \tag{17}$$

Algorithm 2: Optimization Algorithm for (13).

Input: Given data $f(\mathbf{X}), g(\mathbf{Y})$, parameters $\beta_1, \beta_2, \mu_1, \mu_2$, and joint sparsity s_1, s_2

Initialize: $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{D}^0)$

While not converged **do**

1: Update $\mathbf{A}^{k+1}$
2: Update $\mathbf{B}^{k+1}$
3: Update $\mathbf{C}^{k+1}$
4: Update $\mathbf{D}^{k+1}$
5: Check convergence

End while

Output: $(\mathbf{A}^{k+1}, \mathbf{B}^{k+1}, \mathbf{C}^{k+1}, \mathbf{D}^{k+1})$

In order to analyze this problem, the matrix norm is expanded as follows:

$$\frac{1}{2}\mathrm{Tr}(\mathbf{C}^T\mathbf{C}) - \mathrm{Tr}(\mathbf{C}^T(f(\mathbf{X})^T\mathbf{A} + \mathbf{D}^k/\mu_1)) + \frac{1}{2}\mathrm{Tr}((f(\mathbf{X})^T\mathbf{A}+\mathbf{D}^k/\mu_1)^T(f(\mathbf{X})^T\mathbf{A}+\mathbf{D}^k/\mu_1)). \tag{18}$$

Since the first item is constant and the last item is not associated with $\mathbf{C}$, thus only the middle item needs to be maximized. Let the singular value decomposition of $f(\mathbf{X})^T\mathbf{A} + \mathbf{D}^k/\mu$ be $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$. Thus, the middle item can be formulated as

$$\mathrm{Tr}(\mathbf{C}^T(f(\mathbf{X})^T\mathbf{A} + \mathbf{D}^k/\mu)) = \mathrm{Tr}(\hat{\mathbf{C}}^T\mathbf{U}\mathbf{\Lambda}) \tag{19}$$

where $\hat{\mathbf{C}} = \mathbf{C}\mathbf{V}$. Invoking the fact that $\mathbf{V}$ is orthonormal, it is not hard to check that $\hat{\mathbf{C}}^T\hat{\mathbf{C}} = \mathbf{V}^T\mathbf{C}^T\mathbf{C}\mathbf{V} = \mathbf{I}$. Moreover, $\mathbf{\Lambda}$ is diagonal, hence the maximum (minus minimum) is achieved when the diagonal of $\hat{\mathbf{C}}^T\mathbf{U}$ is positive and maximized. This, together with the Cauchy–Schwartz inequality, implies that the optimal solution is obtained when $\hat{\mathbf{C}} = \mathbf{U}$, which yields the solution $\mathbf{C}^{k+1} = \mathbf{U}\mathbf{V}^T$.

Therefore, the proposed DCCA-SCO framework can be computed efficiently via Algorithm 2.

B. Monitoring Strategy

An offline modeling and online monitoring procedure will be discussed in this section. In DCCA-SCO, (f, g) are the learned nonlinear projections by the encoder module, thus $\mathbf{A}^T f$ and $\mathbf{B}^T g$ are the final achieved data for modeling. One proposal for CCA-based PM is that it may be much more difficult or inexact to estimate the fault from singular set of process variables, whereas it may be easier to evaluate using two sets of process variables.

Let $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$ be the monitoring samples. Following the arguments as stated in [8], the residual signal $\mathbf{r}$ can be generated as

$$\mathbf{r} = \mathbf{A}^T f(\mathbf{x}) - \mathbf{\Sigma}\mathbf{B}^T g(\mathbf{y}) \tag{20}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$ is the correlation matrix of $\mathbf{A}^T f$ and $\mathbf{B}^T g$. Although the T^2 statistic and Q statistic can be used to monitor changes of the generated residuals, however, as discussed in [8]–[12], if there is no available prior information about the fault, the T^2 statistic achieves the best detectability for CCA-based PM methods. Thus, the corresponding T^2 test statistical metric is estimated by

$$T^2 = \mathbf{r}^T(\mathbf{I} - \mathbf{\Sigma}^2)^{-1}\mathbf{r}. \tag{21}$$

In addition, the associated control limit or threshold can be chosen by the following $\mathcal{X}^2$ distribution, that is

$$J_{th} = \mathcal{X}^2_\alpha(r) \tag{22}$$

which relies on the degrees of freedom r and the significance level α.

Naturally, when the T^2 statistic violates the aforementioned control limit, a fault has occurred, otherwise there exists no fault. Therefore, the fault detection logic can be given by

$$\begin{cases} T^2 > J_{th} & \Rightarrow \quad \text{faulty} \\ T^2 \le J_{th} & \Rightarrow \quad \text{fault-free} \end{cases}. \tag{23}$$

To sum up, the fault detection strategy consists of offline modeling and online monitoring, as presented in Fig. 3.

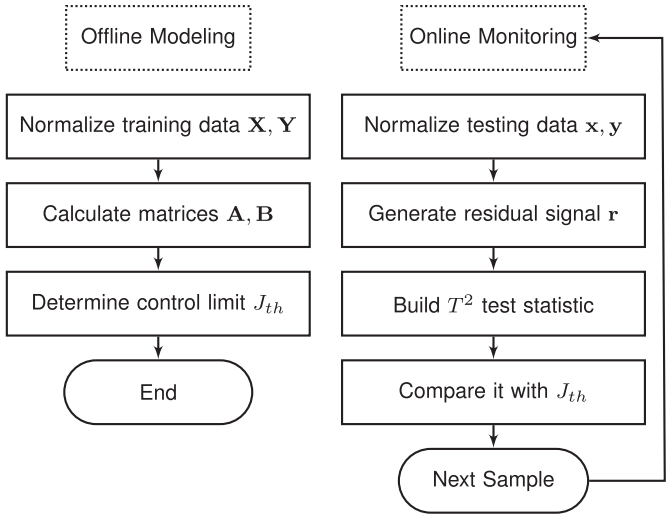


Fig. 3. Fault detection strategy.

IV. Application to the TE Process

This section shows the advantages of the proposed DCCA-SCO compared to CCA [8], CCA-SCO, KCCA [17], KCCA-SCO, and DCCA. In particular, CCA-SCO and KCCA-SCO are developed based on Chen *et al.* [8] and Liu *et al.* [17] by adding the SCO, and DCCA is implemented by integrating CCA and DAENNs. It is worth pointing out that Ma *et al.* [21] and Ren *et al.* [22] are not compared because this article only focuses on the CCA-based methods.

In particular, Section IV-A gives an introduction to the TE benchmark process. Section IV-B describes the experimental setup and measurement indices. Section IV-C presents the monitoring results and shows the advantages based on three specific faults. Section IV-D discusses some remaining issues.

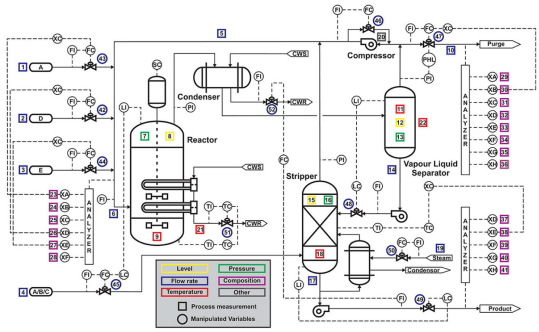


Fig. 4. Flowchart of the TE process.

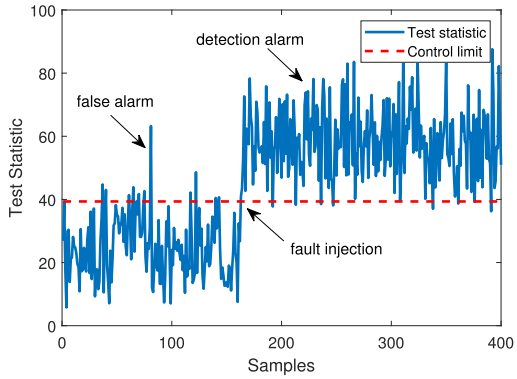


Fig. 5. Definition of different detection indices.

A. Data Description

The TE process is an open-source benchmark industrial process, which is extensively used for validating different PM methods. See Fig. 4 for an overview and more detailed information can be found in [33]. It has five units, i.e., the reactor, the condenser, the separator, the stripper, and the compressor. In the process, two variable sets, i.e., XMV and XMEAS, are collected. The XMV set contains 11 operation variables, and the XMEAS set contains 41 measurement variables. For all the compared methods, XMV(1-11) are selected as $\mathbf{X}$, and XMEAS(1-22) are selected as $\mathbf{Y}$.

In the TE process, 22 training datasets and 22 testing datasets are simulated. Among them, one testing dataset is obtained under normal operational conditions, and the other 21 datasets are collected under 21 different faulty conditions. Each training dataset records 1168 fault-free samples, whereas each validation dataset records 292 fault-free samples. For each faulty dataset, 960 samples are recorded and a fault is injected at the 161st sample. These faults can be divided into five categories: the step change, the random variation, the slow drift, the sticking, and the unknown. In addition, based on the arguments given by Chiang *et al.* [7], fault IDV (3), IDV (9), IDV (15), and IDV (21) are relatively difficult to detect using data-driven monitoring methods. As a result, these faults are not considered here.

B. Setup

In the experiments, parameters $\beta_1, \beta_2 > 0$ and $\mu_1, \mu_2 > 0$ are chosen by fivefold cross-validation with the candidate set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. In addition, the hyperparameters s_1 and s_2 should be chosen carefully. Larger s_1 and s_2 make the constraints meaningless, whereas smaller s_1 and s_2 derive bad representations. Thus, s_1 and s_2 are initialized with small values and then increased gradually by a proper ratio. Based on experience, the significance level α is chosen as 0.05.

For kernel-based PM methods, the Gaussian kernel function is set as

$$\kappa_{\mathrm{X}}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{c}}$$

where c is a kernel bandwidth parameter. It is suggested that $c = \tau p \sigma^2$, in which p is the number of variables, σ^2 is the variance of the monitoring variables, and τ is chosen as 10.

The network contains the encoder module, the CCA module, and the decoder module, where the encoder and the decoder are both constructed for feature selection. For DAENN-based models, the starting layer units are the dimensions of variables, i.e., 11 and 22. The nonlinear projections (f, g) are implemented by networks of two hidden layers, each of which has 11 units and 22 units, respectively. The output layer units are chosen as s_1 and s_2 defined before. In turn, the nonlinear projections (p, q) are also implemented by networks of two hidden layers. The three modules can be optimized using the stochastic gradient descent algorithm with a learning rate 0.01 and momentum 0.99. Furthermore, a small weight decay parameter of 10^{-4} is used for all layers. For the activation function, ReLu is used to introduce nonlinearity as

$$\mathrm{ReLu}(x) = \begin{cases} x, & x > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{24}$$

To measure the detection performance, two indices are introduced, including fault detection rate (FDR) and false alarm rate (FAR). A fault detection sample is defined as a sample that is detected when a fault occurs ($f \neq 0$), whereas a false alarm sample is defined as a sample that is alarmed when there is no fault ($f = 0$). Fig. 5 draws an intuitive diagram, in which a fault is injected at the 161st sample and a fault detection sample and a false alarm sample are marked with arrows. From the perspective of probability, FDR and FAR can be defined as the probability of fault detection samples and false alarm samples, which are given by

$$\begin{aligned} \mathrm{FDR} &= \mathrm{prob}(T^2 > J_{th,T^2} \mid f \neq 0) \\ \mathrm{FAR} &= \mathrm{prob}(T^2 > J_{th,T^2} \mid f = 0). \end{aligned} \tag{25}$$

It can be seen that the aforementioned FDR and FAR values range from [0,1], where a higher FDR or a lower FAR implies better fault detection performance.

C. Monitoring Results

Table I lists the computational results for the selected faults in the TE process. According to this table, some conclusions can be easily obtained, which are as follows.

TABLE I
DETECTION RESULTS IN TERMS OF FDR AND FAR

Fault No.	CCA		CCA-SCO		KCCA		KCCA-SCO		DCCA		DCCA-SCO	
	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR
IDV(1)	99.75%	0.63%	99.75%	0.63%	99.88%	0.63%	99.88%	0.00%	99.88%	0.00%	99.88%	0.00%
IDV(2)	96.50%	0.63%	97.25%	0.63%	98.38%	0.00%	98.38%	0.00%	98.47%	0.00%	99.50%	0.00%
IDV(4)	100%	1.88%	100%	1.25%	100%	1.25%	100%	0.63%	100%	0.63%	100%	0.00%
IDV(5)	100%	3.75%	100%	3.25%	100%	2.50%	100%	2.50%	100%	2.50%	100%	1.88%
IDV(6)	100%	4.38%	100%	4.38%	100%	3.75%	100%	3.25%	100%	3.25%	100%	3.25%
IDV(7)	100%	3.75%	100%	3.25%	100%	2.50%	100%	1.75%	100%	1.25%	100%	0.63%
IDV(8)	96.50%	1.88%	97.50%	1.88%	97.85%	0.63%	98.25%	0.63%	98.88%	0.63%	99.38%	0.00%
IDV(10)	86.88%	1.25%	87.75%	0.63%	89.58%	0.63%	89.75%	0.63%	90.38%	0.00%	93.88%	0.00%
IDV(11)	76.50%	0.63%	77.50%	0.63%	78.50%	0.63%	79.63%	0.63%	80.13%	0.63%	84.50%	0.00%
IDV(12)	99.00%	1.25%	99.00%	0.63%	99.25%	0.00%	99.37%	0.00%	99.50%	0.63%	99.75%	0.00%
IDV(13)	95.75%	0.63%	96.13%	0.63%	96.50%	0.63%	96.50%	0.63%	96.75%	0.00%	96.88%	0.00%
IDV(14)	100%	1.88%	100%	1.25%	100%	0.63%	100%	0.63%	100%	0.63%	100%	0.63%
IDV(16)	93.00%	7.50%	94.38%	5.63%	95.63%	1.25%	96.63%	1.25%	96.63%	1.25%	98.75%	0.63%
IDV(17)	94.13%	3.13%	94.13%	2.50%	94.25%	2.50%	95.13%	1.75%	95.75%	1.25%	96.38%	1.25%
IDV(18)	90.88%	1.88%	91.25%	1.25%	92.50%	0.00%	92.75%	0.00%	93.50%	0.63%	95.63%	0.00%
IDV(19)	92.00%	1.25%	92.63%	1.25%	94.25%	1.25%	94.25%	0.63%	94.93%	0.63%	95.50%	0.63%
IDV(20)	86.88%	0.63%	87.13%	0.63%	87.75%	0.63%	87.75%	0.00%	88.88%	1.25%	89.38%	0.00%

1) Nonlinear CCA methods, such as KCCA, KCCA-SCO, DCCA, and DCCA-SCO, perform better than or as good as CCA and CCA-SCO in terms of FDR and FAR. In particular, for fault IDV(10), the gains of FDR values are 2.70%, 2.87%, 3.50%, and 7.00% for KCCA, KCCA-SCO, DCCA, and DCCA-SCO, respectively, whereas the FAR values are similar.
2) Unlike KCCA and KCCA-SCO, the DCCA and DCCA-SCO can learn the nonlinear structures of the process data by DNNs, which provides an advantage compared with KCCA and KCCA-SCO. For fault IDV(8), the FDR value of DCCA is 1.03% higher than that of KCCA, and 2.38% higher than that of KCCA-SCO, whereas the FDR value of DCCA-SCO is 1.47% higher than that of KCCA, and 2.88% higher than that of KCCA-SCO.
3) For all the mentioned faults, the proposed DCCA-SCO can achieve better results than others, i.e., CCA, CCA-SCO, KCCA, KCCA-SCO, and DCCA.

To verify the monitoring performance achieved by the proposed DCCA-SCO, the visual results are shown through three typical faults, i.e., IDV(4), IDV(10), and IDV(19). Fault IDV(4) is a step change, which is easy to detect. Fig. 6 presents the monitoring performance for all the methods. It can be seen that all six methods can recognize it at the 161st sample and detect it continuously. Both kernel- and DNN-based methods can lift the magnitude, which makes the fault easier to detect. It indicates that nonlinear monitoring is promising and necessary even for easy tasks. However, it is difficult to say that neither of the nonlinear methods is clearly superior to the other.

Fault IDV(10) involves a random variation, which results in a change of the stripper and condenser conditions. Fig. 7 shows the T^2 statistic for all the methods. In comparison, DCCA and DCCA-SCO have very outstanding detection results, i.e., the computed values between the 400th and 600th samples, which are highlighted by the arrows. Moreover, the false alarms are reduced to some degree. This suggests that the DAENNs have advantages compared to linear- or kernel-based PM methods.

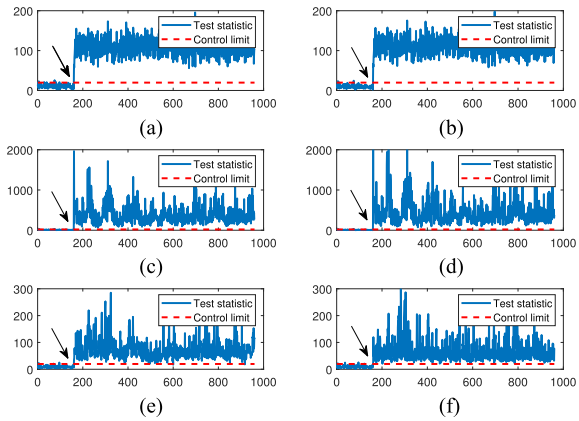


Fig. 6. Monitoring results for fault IDV(4) in the TE process. (a) CCA. (b) CCA-SCO. (c) KCCA. (d) KCCA-SCO. (e) DCCA. (f) DCCA-SCO.

Fault IDV(19) is an unknown fault, which is not easy to detect. Fig. 8 reports the corresponding monitoring performance. Note that DCCA-SCO achieves a better detection capacity than DCCA. Taking a closer look at the values around the 400th sample, the test statistic of DCCA-SCO is larger than these of CCA, CCA-SCO, KCCA, KCCA-SCO, and DCCA. It proves that the SCO is also important for fault detection.

D. Discussions

1) Convergence Verification: Fig. 9 plots the loss function for fault IDV(4) when the number of iterations increases. Since the goal of DCCA-SCO is to minimize the minus canonical correlations (maximize canonical correlations), the objective values are negative. It can be seen that the proposed DCCO-SCO converges after finite iterations.

2) Detection Time: The average detection time is provided in Table II. Obviously, compared with CCA and CCA-SCO, nonlinear PM methods need more detection time due to the kernel

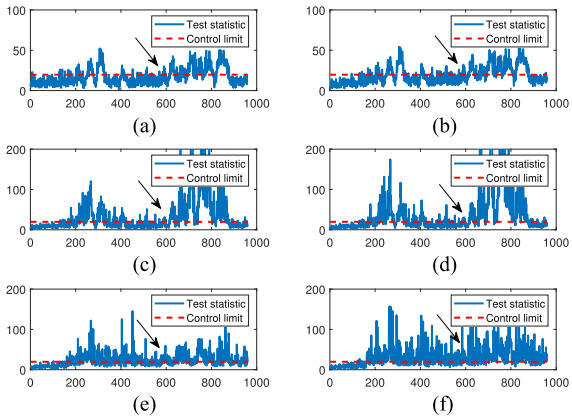


Fig. 7. Monitoring results for fault IDV(10) in the TE process. (a) CCA. (b) CCA-SCO. (c) KCCA. (d) KCCA-SCO. (e) DCCA. (f) DCCA-SCO.

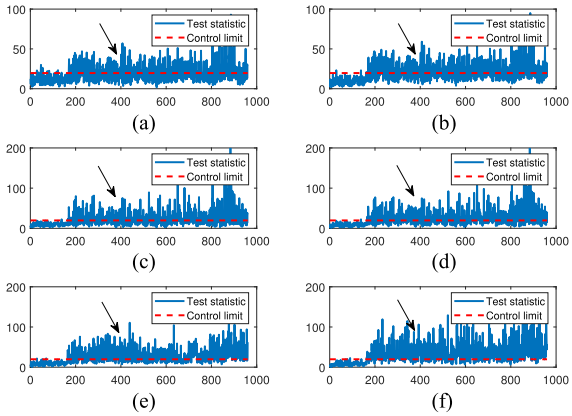


Fig. 8. Monitoring results for fault IDV(19) in the TE process. (a) CCA. (b) CCA-SCO. (c) KCCA. (d) KCCA-SCO. (e) DCCA. (f) DCCA-SCO.

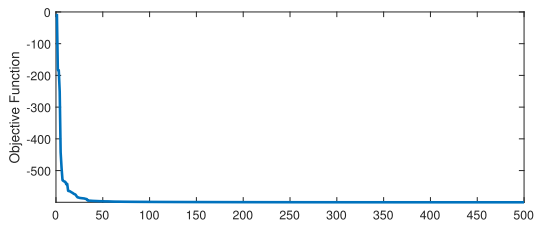


Fig. 9. Convergence of objective values.

TABLE II
DETECTION TIME IN SECONDS

CCA	CCA-SCO	KCCA	KCCA-SCO	DCCA	DCCA-SCO
0.044	0.072	15.676	22.556	2.806	3.097

computing and network training. Although KCCA and KCCA-SCO can also capture the nonlinear structures effectively, a long time is often required. However, DCCA and DCCA-SCO can be trained efficiently by backpropagation. Even though DCCA-SCO involves an iterative hard thresholding computation, it does not take much more time.

TABLE III
MONITORING PERFORMANCE UNDER DIFFERENT LAYERS

Hidden layers	1	2	3
FDR	89.75%	93.88%	93.88%
FAR	0.63%	0.00%	0.00%

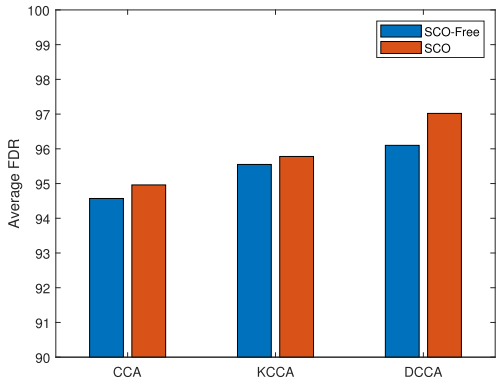


Fig. 10. Illustrations of the improvement.

3) Layer Selection: Table III shows the achieved FDR and FAR values for fault IDV(10) under different hidden layers. It is found that when the number of hidden layers is 2, the best detection result is obtained.

4) Improvement Analysis: Since the proposed DCCA-SCO integrates the CCA, DAENNs, and SCO, the reasons behind the improvement should be distinguished. Fig. 10 shows the average FDR values by different PM methods, where blue indicates SCO-free ones, i.e., CCA, KCCA, and DCCA, and orange indicates SCO ones, i.e., CCA-SCO, KCCA-SCO, and DCCA-SCO. It can be observed that the SCO is useful for CCA, KCCA, and DCCA with positive increases. This is because the SCO can discard outliers and noises, so as to improve the performance of representation. In addition, DCCA and DCCA-SCO perform better than other linear- and kernel-based methods, which illustrates that DAENNs are helpful for learning nonlinear structures of process data. Although the classical kernel-based methods can also learn the nonlinear structures as DAENNs, more parameters need to be tuned. This brings a huge computational load and, thus, limits the scalability to large-scale applications. Though the good monitoring performance of DCCA-SCO can be attributed to both SCO and DAENNs, one can see from Fig. 10 that the SCO can improve the performance for all CCA, KCCA, DCCA and the increase percentage is much higher in DCCA case. This indicates DCCA, especially DAENNs, plays a more important role in the proposed DCCA-SCO.

V. APPLICATION TO THE DG PROCESS

A. Data Description

The DG is an important part to ensure the normal operation of marine vessels [34]. It has six parts: the oil lubrication system, the fuel system, the control and protection system, the cooling system, the exhaust system, and the starting system. The dataset

TABLE IV
SELECTED VARIABLES IN THE DG PROCESS

Var.	Description
1	oil inlet temperature
2	cooling water outlet temperature
3	exhaust gas into supercharger A temperature
4	exhaust gas into supercharger B temperature
5	high temperature cooling water inlet pressure
6	starting air pressure
7	oil inlet pressure
8	running speed
9	running load
10	operating hours
11	bearing temperature
12	U-phase winding temperature
13	V-phase winding temperature
14	W-phase winding temperature

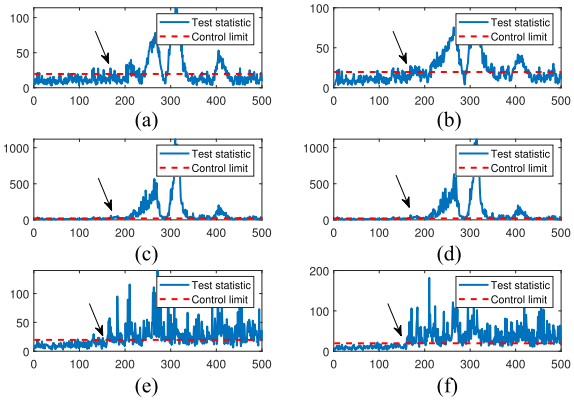


Fig. 11. Monitoring results for the DG process. (a) CCA. (b) CCA-SCO. (c) KCCA. (d) KCCA-SCO. (e) DCCA. (f) DCCA-SCO.

is collected in a real, two-stroke low-speed marine diesel engine, which consists of three generators.

For each DG, 14 variables are selected, as listed in Table IV. Therefore, 42 variables are chosen in total. In this study, variables 1–10 are set as $\mathbf{X}$, and variables 11–14 are set as $\mathbf{Y}$. In the DG process, 500 samples are selected for offline modeling. Moreover, a fault is injected at the 161st sample for testing PM.

B. Monitoring Results

Fig. 11 shows the monitoring results for the DG process. It can be seen that CCA and CCA-SCO cannot detect the fault immediately after the fault is injected at the 161st sample, whereas KCCA, KCCA-SCO, DCCA, and DCCA-SCO can complete the task. Furthermore, DCCA-SCO achieves better monitoring performance than others because almost all the faulty samples that violated the control limit are detected, showing that DCCA-SCO is able to detect the fault consistently. At the same time, DCCA-SCO obtains a very similar percentage of false alarms.

From the aforementioned two numerical studies, it is concluded that the proposed DCCA-SCO framework delivers a more encouraging fault detectability than linear and kernel data-driven PM methods, because DNNs can learn appropriate nonlinear projection functions automatically. In addition, the introduction of SCO is able to improve the representation of process variables.

VI. CONCLUSION

In this article, a new CCA framework was proposed to improve the performance of nonlinear PM. The DAENNs were used to seek a proper nonlinear projection automatically. After that, the CCA-based algorithm was implemented to explore the relationship among process variables. In particular, the SCO was used to enhance the robustness and extract important variables. Both of the DAENNs and SCO contributed to the good performance of the proposed method. To check whether a fault occurred, a residual signal was generated and the T^2 test statistic was constructed. The extensive experimental results on the benchmark TE process and the DG process illustrated its effectiveness. The monitoring results showed that the proposed method could perform better than the classical CCA and KCCA.

Although the proposed DCCA-SCO achieves satisfactory performance for fault detection, several interesting and important questions need to be further investigated. First, as the current work focuses on studying the effectiveness of SCO-aided PM, there is less research on how to select neural networks. Therefore, a comprehensive study of DNNs-SCO-based CCA methods should be analyzed. Second, only fault detection is discussed in this article, the fault isolation and diagnosis are not studied theoretically and numerically. To the best of our knowledge, it is still an open question to apply the CCA-based methods to fault isolation and diagnosis. The reason is that the original process variables cannot be tracked after nonlinear projections. Third, the unsupervised feature learning [35] has been successfully applied in SAR segmentation and it can be used for PM in the future. Finally, despite constructing for process control, the proposed method can also be applied to face recognition and human monitoring.

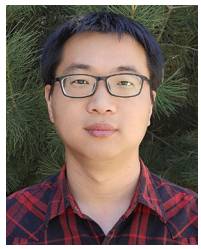
ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and anonymous reviewers for their numerous constructive comments, which have greatly improved this article. They would also like to thank Dr. P. Shang from Beijing Jiaotong University for insightful advices.

REFERENCES

[1] S. Ding, *Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems*. Berlin, Germany: Springer, 2014.
[2] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
[3] Z. Gao and X. Liu, "An overview on fault diagnosis, prognosis and resilient control for wind turbine systems," *Processes*, vol. 9, no. 2, 2021, Art. no. 300.
[4] Y. Liu, J. Zeng, L. Xie, S. Luo, and H. Su, "Structured joint sparse principal component analysis for fault detection and isolation," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2721–2731, May 2019.

[5] Y. Fu, Z. Gao, Y. Liu, A. Zhang, and X. Yin, "Actuator and sensor fault classification for wind turbine systems based on fast Fourier transform and uncorrelated multi-linear principal component analysis techniques," *Processes*, vol. 8, no. 9, 2020, Art. no. 1066.
[6] S. Yin, S. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.
[7] L. Chiang, E. Russell, and R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. Berlin, Germany: Springer, 2000.
[8] Z. Chen, S. Ding, K. Zhang, Z. Li, and Z. Hu, "Canonical correlation analysis-based fault detection methods with application to alumina evaporation process," *Control Eng. Pract.*, vol. 46, pp. 51–58, 2016.
[9] Z. Chen, K. Zhang, S. Ding, Y. Shardt, and Z. Hu, "Improved canonical correlation analysis-based fault detection methods for industrial processes," *J. Process Control*, vol. 41, pp. 26–34, 2016.
[10] Q. Jiang, S. Ding, Y. Wang, and X. Yan, "Data-driven distributed local fault detection for large-scale processes based on the GA-regularized canonical correlation analysis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8148–8157, Oct. 2017.
[11] Q. Jiang and X. Yan, "Multimode process monitoring using variational Bayesian inference and canonical correlation analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1814–1824, Oct. 2019.
[12] B. Song, H. Shi, S. Tan, and Y. Tao, "Multi-subspace orthogonal canonical correlation analysis for quality related plant wide process monitoring," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6368–6378, Sep. 2021.
[13] H. Hotelling, "Relations between two sets of variates," *Econometrica*, vol. 28, no. 3/4, pp. 321–377, 1936.
[14] X. Yang, L. Weifeng, W. Liu, and D. Tao, "A survey on canonical correlation analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2349–2368, Jun. 2021.
[15] H. Akaike, "Stochastic theory of minimal realization," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 667–674, Dec. 1974.
[16] W. Larimore, "Canonical variate analysis in identification, filtering, and adaptive control," in *Proc. 29th IEEE Conf. Decis. Control*, 1990, pp. 596–604.
[17] Q. Liu, Q. Zhu, S. Qin, and T. Chai, "Dynamic concurrent kernel CCA for strip-thickness relevant fault diagnosis of continuous annealing processes," *J. Process Control*, vol. 67, pp. 12–22, 2018.
[18] S. Mehrkanoon and J. Suykens, "Regularized semipaired kernel CCA for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3199–3213, Jul. 2018.
[19] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.
[20] Q. Jiang and X. Yan, "Learning deep correlated representations for nonlinear process monitoring," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6200–6209, Dec. 2019.
[21] M. Ma, C. Sun, and X. Chen, "Deep coupling autoencoder for fault diagnosis with multimodal sensory data," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1137–1145, Mar. 2018.
[22] Z. Ren, W. Zhang, and Z. Zhang, "A deep nonnegative matrix factorization approach via autoencoder for nonlinear fault detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5042–5052, Aug. 2020.
[23] J. Wu, S. Sfarra, and Y. Yao, "Sparse principal component thermography for subsurface defect detection in composite products," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5594–5600, Dec. 2018.
[24] X. Peng, Y. Tang, W. Du, and F. Qian, "Multimode process monitoring and fault detection: A sparse modeling and dictionary learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4866–4875, Jun. 2017.
[25] R. Wang, N. Xiu, and C. Zhang, "Greedy projected gradient-Newton method for sparse logistic regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 527–538, Feb. 2020.
[26] X. Yuan, P. Li, and T. Zhang, "Gradient hard thresholding pursuit," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6027–6069, 2017.
[27] M. Xu, Z. Zhu, X. Zhang, Y. Zhao, and X. Li, "Canonical correlation analysis with $l_{2,1}$-norm for multiview data representation," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4772–4782, Nov. 2020.
[28] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[29] A. Ng, "Sparse autoencoder," *CS294 Lecture Notes*, vol. 72, no. 2011, pp. 1–19, 2011.
[30] X. Xiu, Y. Yang, L. Kong, and W. Liu, "Laplacian regularized robust principal component analysis for process monitoring," *J. Process Control*, vol. 92, pp. 212–219, 2020.
[31] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Delft, The Netherlands: Now Publishers, Inc., 2011.
[32] L. Pan and X. Chen, "Group sparse optimization for images recovery using capped folded concave functions," *SIAM J. Imag. Sci.*, vol. 14, no. 1, pp. 1–25, 2021.
[33] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, 1993.
[34] K. Mollenhauer and H. Tschoke, *Handbook of Diesel Engines*. Berlin, Germany: Springer-Verlag, 2010.
[35] M. Rahmani and G. Akbarizadeh, "Unsupervised feature learning based on sparse coding and spectral clustering for segmentation of synthetic aperture radar images," *IET Comput. Vis.*, vol. 9, no. 5, pp. 629–638, 2015.

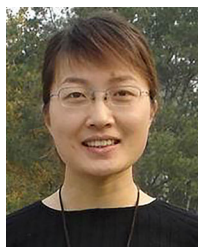


Xianchao Xiu (Member, IEEE) received the Ph.D. degree in operations research from Beijing Jiaotong University, Beijing, China, in 2019.

From June 2019 to May 2021, he was a Postdoctoral Researcher with Peking University, Beijing. He is currently a Faculty Member with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China. His current research interests include large-scale sparse optimization, signal processing, deep learning, and data-driven fault detection.

Zhonghua Miao received the Ph.D. degree in mechatronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010.

He is currently a Full Professor and Doctoral Advisor with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai. His research interests include intelligent robotics, measurement and control, fault diagnosis, and agricultural machinery equipment.



Ying Yang (Senior Member, IEEE) received the Ph.D. degree in control theory from Peking University, Beijing, China, in 2002.

From January 2003 to November 2004, she was a Postdoctoral Researcher with Peking University. From 2005 to 2014, she was an Associate Professor with the Department of Mechanics and Engineering Science, College of Engineering, Peking University, where she has been a Full Professor since 2014. Her current research interests include robust and optimal control, nonlinear systems control, numerical analysis, fault detection, and fault tolerant systems.

Wanquan Liu (Senior Member, IEEE) received the B.S. degree in applied mathematics from Qufu Normal University, Jining, China, in 1985, the M.S. degree in control theory and operation research from the Chinese Academy of Science, Beijing, China, in 1988, and the Ph.D. degree in electrical engineering from Shanghai Jiaotong University, Shanghai, China, in 1993.

He is currently a Full Professor with the School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, China. He once held the ARC Fellowship, U2000 Fellowship, and JSPS Fellowship and attracted research funds from different resources over 2.4 million dollars. His current research interests include large-scale pattern recognition, signal processing, machine learning, and control systems.