Chemical Engineering Science 72 (2012) 78–86

Contents lists available at SciVerse ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Dynamic processes monitoring using recursive kernel principal component analysis

Yingwei Zhang [a,b,*], Shuai Li [a,b], Yongdong Teng [a,b]

[a] State Key Laboratory of Synthetical Automation of Process Industry, Northeastern University, Shenyang, Liaoning 110819, PR China
[b] Key Laboratory of Advanced Control and Optimization for Chemical Processes, Shanghai 200237, PR China

ARTICLE INFO

Article history:
Received 30 April 2011
Received in revised form
4 November 2011
Accepted 20 December 2011
Available online 3 January 2012

Keywords:
Time-varying
Nonlinear processes
Fault detection
Recursive principal component analysis
Design
Control

ABSTRACT

The dynamic process monitoring is discussed in this paper. Kernel principal component analysis (KPCA) is a nonlinear monitoring method that cannot be employed for dynamic systems. Recursive KPCA (RKPCA) is proposed to monitor the dynamic processes, which is adaptive monitoring method by computing recursively the eigenvalues and eigenvectors in the kernel space when the training data are updated dynamically. The contributions of this article are as follows: (1) The model of history data is used to build new model after the new sample is obtained. The expensive computation is avoided in this article. (2) New nonlinear modeling method is proposed based on a new singular value decomposition (SVD) technique. The results are interesting due to the nonlinear time evolution of the variables involved. The proposed algorithm was applied to the continuous annealing process and penicillin fermentation process for adaptive monitoring and RKPCA could efficiently capture the time-varying and nonlinear relationship in process variables.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The demand for safe operation in the industry has propelled research into fault detection and diagnosis methods. Multivariate statistical methods such as principal component analysis (PCA) (AlGhazzawi and Lennox, 2008; Wang et al., 2005; Chiang et al., 2001), partial least squares(Kruger et al., 2001; Morud, 1996; Zhang et al., 2006) and more recently independent component analysis (Ge and Song, 2007; Lee et al., 2006; Kano et al., 2003) have been investigated for process monitoring. Among them, PCA is the most popular one, which relates to its conceptual simplicity. However, PCA methods assume linear variable relationships in the process, which limit their application if these relationships are nonlinear.

Kramer(1991) developed a nonlinear PCA, which relies on auto-associative neural networks (ANNs). Dong and McAvoy proposed a simplification that incorporated principal curves into this neural network structure (Dong and McAvoy, 1996). The nonlinear variable relationship was addressed by introducing input-training neural networks (Jia et al., 2000; Tan and Mavrovouniotis, 1995). An excellent review is given in Kruger et al. (2007), which suggests that neural network cannot be used to represent a generic nonlinear extension. KPCA (Scholkopf et al., 1999; Zhang and Qin, 2007, 2008; Jeng et al., 2007; Voegtlin, 2005) is a generic nonlinear PCA extension, which can efficiently compute PCs in a high-dimensional feature space using nonlinear kernel functions. The core idea of KPCA is to first map the data space into a feature space using a nonlinear mapping and then compute the PCs in the feature space. It should also be noted that KPCA only requires the solution of an eigenvalue problem, and, since it can incorporate different kernel functions, KPCA can handle a wide range of nonlinearities. However, a major limitation of KPCA-based monitoring is that the KPCA model is time-invariant. Most real industrial processes are time-varying (Gallagher et al., 1997). The time-varying characteristics of industrial processes include (1) changes in the mean; (2) changes in the variance; and (3) changes in the correlation structure among variables, including changes in the number of significant principal components (PCs). When a static KPCA model is used to monitor processes with the dynamic changes, false alarms often occur, which significantly reduce the reliability of the method. A recursive algorithm for PCA based on rank-one matrix update of the covariance is derived to compute the LPC on line (Zhang and Qin, 2007; Jeng et al., 2007). A recurrent linear network was applied to RPCA for monitoring purpose (Zhang and Qin, 2008; Voegtlin, 2005). Recursive partial least squares (RPLS) algorithms together with adaptive confidence limits can lead to a considerable reduction in the number of false alarms (Wang et al., 2003). In Liu et al. (2009), the proposed technique incorporates an up- and downdating procedure to adapt the data mean and

* Corresponding author.
E-mail addresses: zhangyi@che.utexas.edu, zhangyingwei@mail.neu.edu.cn (Y. Zhang).

0009-2509/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ces.2011.12.026

covariance matrix in the feature space. The eigenvalues and eigenvectors of the Gram matrix were approximated. The adaptation mechanism is based on the incremental method proposed by Hall et al. (Elshenawy et al., 2010; Hall et al., 2000). A fault detection tool based on Kernel PCA is tested in multiple operation modes environments (Maestri et al., 2009). Conventional kernel principal component analysis (KPCA) may not function well for nonlinear processes, since the Gaussian assumption of the method may be violated through nonlinear and kernel transformation of the original process data. To overcome this deficiency, a statistical local approach is incorporated into KPCA (Ge et al., 2009). The multivariate exponentially moving average (MEWMA) was applied to capture the time-varying process shifts and then incorporating with KPCA components to develop an adaptive monitoring statistic (Cheng et al., 2010). Two RPCA algorithms were proposed to greatly reduce the computation cost (Elshenawy et al., 2010). The first algorithm is based on first-order perturbation analysis (FOP), which is a rank-one update of the eigenvalues and their corresponding eigenvectors of a sample covariance matrix. The second one is based on the data projection method (DPM), which is a simple and reliable approach for adaptive subspace tracking. However, the rank-one update is not available for the kernel method since the mapping function is unknown. The differences between the proposed method and the existing work in this article are as follows:

(1) In the existing work, the history data and the new sample form an entirety. New model was re-built using the existing method after the new sample was obtained. However, the model of history data was given up when the new model was built. In this article, the model of history data is used to build new model based on new singular value decomposition (SVD) technique. The expensive computation is avoided and new modeling method is proposed in this article.
(2) In the existing recursive methods, only linear methods were proposed. Because the kernel function is unknown, it is difficult to describe the nonlinear dynamic data structure. In this article, the nonlinear modeling method is proposed based on new singular value decomposition (SVD) technique.

The contributions of the work are as follows:

(1) New singular value decomposition technique is proposed.
(2) The model of history data is used to build new model after the new sample is obtained. The expensive computation is avoided in this article.
(3) New nonlinear modeling method is proposed based on a new singular value decomposition (SVD) technique. The results are interesting due to the nonlinear time evolution of the variables involved.

The paper is organized as follows. Preliminaries of KPCA-based monitoring are presented in Section 2. This is followed by the adaptation of the RKPCA model in Section 3. Section 4 presents the adaptation of the number of retained PCs. Next, the illustrative examples are given to demonstrate the effectiveness of the RKPCA in Section 6. Finally, conclusions are drawn.

2. KPCA algorithm

KPCA is an extension of PCA and it can always be solved as an eigenvalues problem of its Kernel matrix. The use of Nonlinear Iterative Partial Least Squares (NIPALS) can get the same result as the eigenvalue method proposed by Schölkopf et al. (1998). NIPALS gets the PC one by one (Zhang and Ma, 2011; Zhang et al., 2010; Zhang et al., 2010). This algorithm is listed in Table 1. Radial basis function is selected to building the kernel matrix in this paper. The ith row jth column calculated by $K_{i,j}=\exp(-\|\mathbf{x}_i,\mathbf{x}_j\|/c)^2$. In this function, the Euclidean distance of two input vectors are taken as the parameter of exponential (2-norm). Both T^2 statistic and SPE statistic can be used for monitoring the process.

Table 1
NIPALS for KPCA.

	For comprehension	For computation
1	Scale $\mathbf{K}$	Scale $\mathbf{K}$
2	Initialize $\mathbf{t}_i$	Initialize $\mathbf{t}_i$
3	$\mathbf{q}_i=\boldsymbol{\Phi}_i^T\mathbf{t}_i/\|\boldsymbol{\Phi}_i^T\mathbf{t}_i\|$ $\mathbf{t}_i=\boldsymbol{\Phi}_i\mathbf{p}_i$	$\mathbf{t}_i=\mathbf{K}_i\mathbf{t}_i/\sqrt{\mathbf{t}_i^T\frac{\mathbf{K}_i}{N}\mathbf{t}_i}$
	Loop until $\mathbf{t}_i$ converges	Loop until $\mathbf{t}_i$ converges
4	$\boldsymbol{\Phi}_{i+1}=(\mathbf{I}-\mathbf{t}_i\mathbf{t}_i^T/\mathbf{t}_i^T\mathbf{t}_i)\boldsymbol{\Phi}_i$ Go to step 3	$\mathbf{K}_{i+1}=(\mathbf{I}-\mathbf{t}_i\mathbf{t}_i^T/\mathbf{t}_i^T\mathbf{t}_i)\mathbf{K}_i(\mathbf{I}-\mathbf{t}_i\mathbf{t}_i^T/\mathbf{t}_i^T\mathbf{t}_i)$ Go to step 3

Let $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]\in R^{m\times N}$ be the sample matrix. For a new test sample $\mathbf{X}_{new}|\in R^N$, compute the kernel vector

$$\mathbf{k}_{new}=\boldsymbol{\Phi}(\mathbf{x}_{new})\boldsymbol{\Phi}(\mathbf{X})^T\in\mathbf{R}^N \tag{1}$$

Let $\mathbf{A}=[\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_N]\in R^{N\times N}$ be the coefficient matrix, which is computed by

$$\mathbf{A}_i=\mathbf{t}_i/\sqrt{\mathbf{t}_i(\mathbf{K}/N)\mathbf{t}_i}\in\mathbf{R}^N \tag{2}$$

$\mathbf{A}$ is calculated in the modeling period. It can be used to compute scores of the new test sample

$$\mathbf{t}_{new}=\mathbf{k}_{new}\mathbf{A}\in\mathbf{R}^N \tag{3}$$

Note that in modeling period, $\mathbf{t}_i$ denotes ith column of the whole block scores matrix. But here, $\mathbf{t}_{new}$ is a row vector, means the scores on every axis of one test point. The T^2 statistic can be calculated by

$$T_{new}^2=\mathbf{t}_{new}\boldsymbol{\Lambda}^{-1}\mathbf{t}_{new}^T \tag{4}$$

To compute SPE statistic, there is a need for estimating $\boldsymbol{\Phi}(\mathbf{x}_{new})$. This can be done as

$$\hat{\boldsymbol{\Phi}}(\mathbf{x}_{new})=\mathbf{t}_{new}\mathbf{P}^T=\mathbf{t}_{new}\mathbf{A}^T\boldsymbol{\Phi}(\mathbf{X}) \tag{5}$$

where $\mathbf{P}=\boldsymbol{\Phi}(\mathrm{X})^T\mathbf{A}$.

The computation processes of *SPE* statistic and T^2 statistic can be found in Appendix A.

3. Dynamical recursive KPCA method

In Liu et al. (2009), the proposed technique incorporates an up- and downdating procedure to adapt the data mean and covariance matrix in the feature space. The multivariate exponentially moving average (MEWMA) was applied to capture the time-varying process shifts and then incorporating with KPCA components to develop an adaptive monitoring statistic (Cheng et al., 2010). The RPCA algorithms were proposed to greatly reduce the computation cost based on a rank-one update (Elshenawy et al., 2010). In this section, new singular value decomposition technique is proposed. And then new RKPCA algorithm is proposed based on the proposed singular value decomposition technique.

Let $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$ be the sample matrix. To obtain dynamic relation, $\boldsymbol{X}\otimes\boldsymbol{\beta}=[\mathbf{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_0,\mathbf{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_1,\ldots,\mathbf{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_{k-1}]_{m\times k}$ is defined as the dynamic sample matrix, where

$\boldsymbol{\beta}=\left[\boldsymbol{\beta}_0 \quad \cdots \quad \boldsymbol{\beta}_{k-1}\right]$,$\boldsymbol{\beta}_0=\left[0 \quad 0 \quad \ldots \quad 1\right]^T_{N\times 1}$, $\boldsymbol{\beta}_1=\left[0 \quad \ldots \quad 1 \quad 0\right]^T_{N\times 1}$,

$$\boldsymbol{\beta}_{k-1}=\left[0 \quad \ldots \quad 0 \quad \underbrace{1 \quad 0 \quad \ldots \quad 0}_{k}\right]^T_{N\times 1},$$

$\boldsymbol{\beta}_{N-1}=\left[1 \quad 0 \quad \ldots \quad 0\right]^T_{N\times 1}$. For new singular value decomposition of the dynamic matrices, please see Appendix B. And when recursive computing process is extended to the feature space ($\mathbf{F}$), a new recursive KPCA method is proposed to update the eigenvalue decomposition of the covariance matrix $\mathbf{C}^F$, which can be found in Appendix C.

The adaptive algorithm for the KPCA-based monitoring model can be summarized by the following 7 steps:

(1) Let $\mathbf{X}\otimes\boldsymbol{\beta}=[\boldsymbol{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_0,\boldsymbol{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_1,\ldots,\boldsymbol{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_{k-1}]$ be the sample matrix. Let $\tilde{\mathbf{X}}\otimes\tilde{\boldsymbol{\beta}}=[\mathbf{X}[(1:m,2)]\otimes\boldsymbol{\beta},\ldots,\mathbf{X}[(1:m,N)]\otimes\boldsymbol{\beta}_{k-1}]$ be the intermediate matrix. Assume $\mathbf{x}_{new}$ is a new sample, the updated sample matrix is $\mathbf{X}_{new}=[\tilde{\mathbf{X}}\otimes\boldsymbol{\beta} \quad \mathbf{x}_{new}\otimes\boldsymbol{\beta}]$;
(2) Calculate the mean values $\overline{\Phi}(\mathbf{X}\otimes\boldsymbol{\beta})$, $\overline{\Phi}(\tilde{\mathbf{X}}\otimes\boldsymbol{\beta})$ and $\overline{\Phi}(\mathbf{X}_{new}\otimes\boldsymbol{\beta})$of $\Phi(\boldsymbol{X}\otimes\beta)$, $\Phi(\tilde{\mathbf{X}}\otimes\boldsymbol{\beta})$ and $\boldsymbol{\Phi}(\mathbf{X}_{new}\otimes\boldsymbol{\beta})$;
(3) Compute the decompositions of $\overline{\Phi}(\tilde{\mathbf{X}}\otimes\boldsymbol{\beta})$, $\overline{\Phi}(\mathbf{X}\otimes\boldsymbol{\beta})$ and $\overline{\Phi}(\mathbf{X}_{new}\otimes\boldsymbol{\beta})$ according to Eqs. (21), (24) and (28), respectively;
(4) Compute the eigenvectors $\mathbf{P}_{\overline{\Phi}(\mathbf{X}_{new})}$ of $\overline{\Phi}(\mathbf{X}_{new})$;
(5) For a new sample $\mathbf{x}_{new}$, use the KPCA model to determine the score vector, $\mathbf{t}_{new}$;
(6) Determine the Hotelling's T^2 and residual SPE statistics;
(7) If the statistics for $\mathbf{x}_{new}$ exceed the confidence limits, that is $T^2>T^2_{\beta}$ and/or SPE $>$ SPE$_\beta$, then an alarm occurs.

4. Recursive determination of the number of PCs

Since the number of significant principal components can change over time, it is necessary to determine this number recursively in RKPCA modeling. There are many ways of determining the number of PCs in batch-wise PCA, including

- Cross-validation (Tracy et al., 1992; Nomikos and MacGregor, 1995)
- Cumulative percent variance (Shrager and Hendler, 1982)
- Scree test (Wold, 1978)
- Average eigenvalues
- Imbedded error function (Shrager and Hendler, 1982)
- Xu and Kailath's approach (Kano et al., 2003)
- Akaike information criterion (Osten, 1988; Malinowski, 1991)
- Minimum description length criterion (Malinowski, 1991; Cattell, 1966)
- Variance of reconstruction error (Xu and Kailath, 1994)

However, not all the approaches are suitable for recursive KPCA. For example, the cross-validation approach is not suitable because old data are not representative for the current process. Since the number of dominant principal components may vary in different batches, it is important to determine the number of PCs adaptively. There are some methods to determine the number, including cumulative percent variance, the scree test, average eigenvalues and the variance of reconstruction error (Liu et al., 2009). In this article, cumulative percent variance method is used to determine the number of nonlinear PCs.

The cumulative percent variance (CPV) is a measure of the percent variance captured by the first r PCs:

$$CPV(r)=\frac{\sum_{j=1}^{r}\lambda_j}{\sum_{j=1}^{N}\lambda_j}100\%$$

the number of PCs is chosen when CPV reaches a predetermined limit, say 95%.

5. Results and discussion

5.1. Monitoring for a continuous annealing process

A continuous annealing process is a highly efficient heat treatment process after cold rolling in steel works (Xu and Kailath, 1994). It produces steel strips of high tensile strength and high formability. It has been applied to all over the world for its high reliability, high quality, high productivity and many other merits (Akaike, 1974). The continuous annealing line continuously heats and cools the strip to make the internal crystal suffer renewal, recrystallization, growth and carbide precipitation in order to improve the reprocessing performance of the strip. The material for annealing is a cold-rolled strip coil, which is put on a pay-off reel on the entry side of the line. The head end of the coil is then pulled out and welded with the tail end of the preceding coil. Then the strip runs through the process with a certain line speed. On the delivery side, the strip is cut into a product length by a shear machine and coiled again by a tension reel.

The physical layout of the continuous annealing process is shown. The maximum line speed is 880 m/min, the width of the strip is 900–1230 mm, the thickness is 0.18–0.55 mm, the maximum weight is 26.5 t and it is heated to 710 °C. The first entry coil is opened by payoff reel (POR), and it is welded into a strip finally. The strip passes through 1# bridle roll (1BR), entry loop (ELP), 2# bridle roll (2BR), 1# dancer roll (1DCR) and 3# bridle roll (3BR), then it enters the continuous annealing furnace. The annealing technologies consist of: rapid cooling–reheating-inclined over ageing. The annealing equipments include heating furnace (HF), soaking furnace (SF), slow cooling furnace (SCF), 1# cooling furnace (1C), reheating furnace (RF), over ageing furnace (OA) and 2# cooling furnace (2C). After completing the annealing craft, the strip in turn passes through 4# bridle roll (4BR), delivery loop (DLP) and 5# bridle roll (5BR), and temper rolling machine (TPM), 6# bridle roll (6BR), 2# dancer roll (2DCR) and 7# bridle roll (7BR). Finally, the strip enters roll type reel (TR) to become coil.

The continuous annealing process is a complex nonlinear and time-varying processes. In this section, KPCA and RKPCA methods are applied to the continuous annealing process for detecting the looper fault. There are 200 samples for modeling. Also, there are 300 samples for test. Each sample has 77 variables. Build the model by KPCA and RKPCA, and the 99% control limits are calculated in each simulation. When the KPCA model is used to monitor the process and not updated with new test data, the calculated T^2 and SPE are shown in Figs. 1 and 2. At the beginning, the continuous annealing process is unstable, so the T^2 and SPE exceed their respective confidence limits for the test data. But for the test data, the looper fault is not detected by the KPCA. Compared to KPCA, RKPCA uses the 200 samples to build the initial RKPCA model, then the model is updated using the recursive singular value decomposition (SVD) in the kernel space. The monitoring results of RKPCA are shown in Figs. 3 and 4. During the recursive updating, the beginning test data, outside the T^2 and/or SPE limit, is not used to update. In Figs. 3 and 4, both T^2 and SPE plots generated by RKPCA can detect the looper fault from about the 175th sample. The results show that RKPCA efficiently capture the time-varying trend in process variables and significantly reduce missing detection rate. Notice that in Figs. 3 and 4 the control limits are also updated based on the available test data.

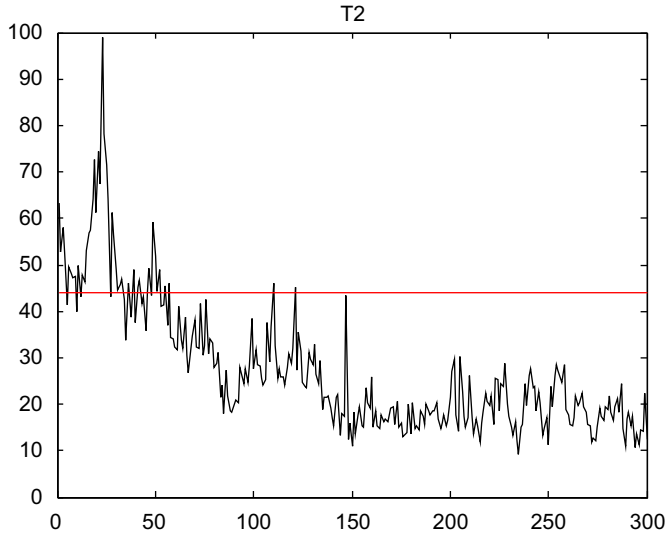

Fig. 1. KPCA monitoring results of the continuous annealing process.

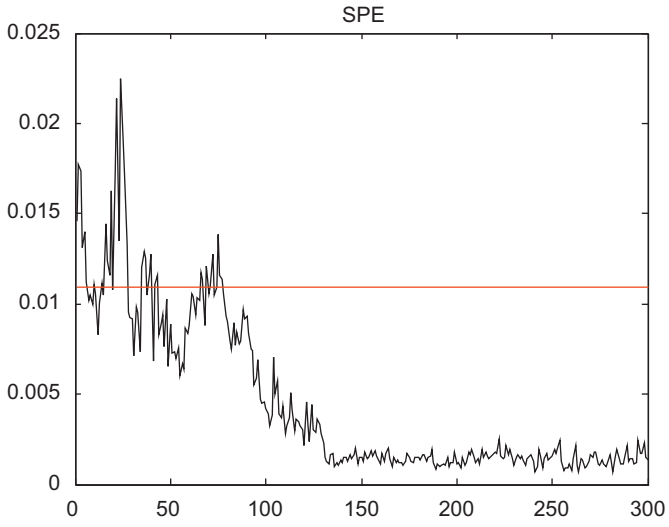

Fig. 2. KPCA monitoring results of the continuous annealing process.

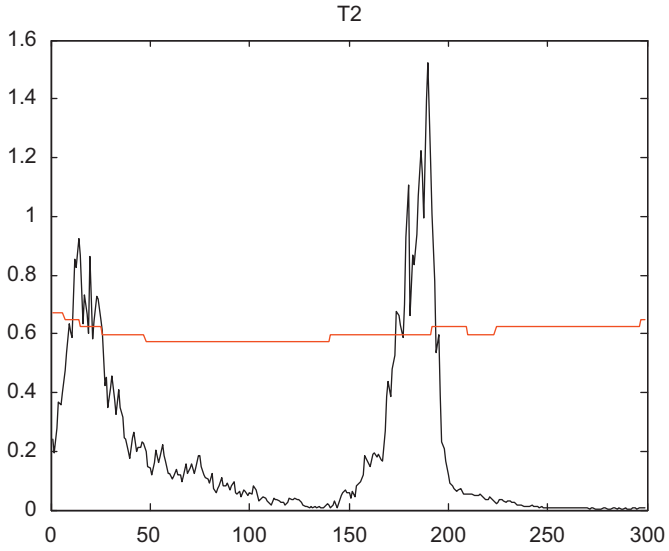

Fig. 3. RKPCA monitoring results of the continuous annealing process.

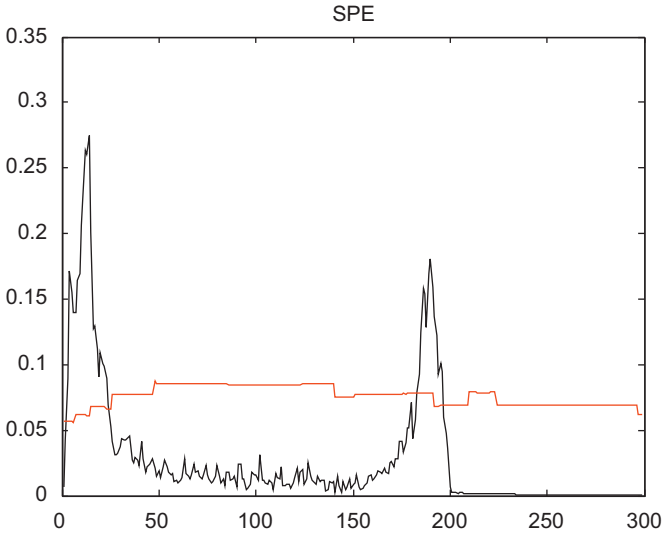

Fig. 4. RKPCA monitoring results of the continuous annealing process.

5.2. *Penicillin fermentation*

In this section, RKPCA method is applied to the monitoring of a well-known benchmark process (Zhang and Qin, 2007; Jeng et al., 2007; Gajardo et al., 2011; Johnson, 1987), penicillin fermentation process. A flow diagram of the penicillin fermentation process is given in Fig. 5. Trajectories of nine variables from a nominal batch run are shown in Fig. 6. The production of secondary metabolites such as antibiotics has been the subject of many studies because of its academic and industrial importance. Here, we focus on the process to produce penicillin, which has non-linear dynamics and multiphase characteristics. In typical operating procedure for the modeled fed-batch fermentation, most of the necessary cell mass is obtained during the initial pre-culture phase. When most of the initially added substrate has been consumed by the microorganisms, the substrate feed begins. Penicillin starts to be generated at the exponential growth phase and continues to be produced until the stationary phase. A low substrate concentration in the fermentor is necessary for achieving a high product formation rate due to the catabolite repressor. Consequently, glucose is fed continuously during fermentation at the beginning. In the present simulation experiment, a total of 60 reference batches are generated using a simulator (PenSim v2.0 simulator). Detail process description is well explained from http://www.chee.iit.edu/~cinar/software.htm. These simulations are run under closed-loop control of pH and temperature, while glucose addition is performed open-loop. Small variations are automatically added to mimic the real normal operating conditions under the default initial setting conditions. The duration of each batch is 400 h, consisting of a pre-culture phase of about 45 h and a fed-batch phase of about 355 h. In the simulation, we choose 400 samples for modeling and 600 samples for test. For the test data, fault is implemented by introducing a 10% step increase in the substrate feed rate at the 200th sample and retaining until the 250th sample. Build the initial RKPCA model, then the model is updated using the test data and the 99% control limits are calculated. The monitoring results of RKPCA are shown in Figs. 7 and 8. In Figs. 7 and 8, both T^2 and SPE plots generated by RKPCA can detect the fault from the 200th sample. The results show that RKPCA efficiently captures the non-linear dynamics in process variables and significantly reduces missing detection rate. Notice that in Figs. 9 and 10 the control limits are also updated based on the available test data.

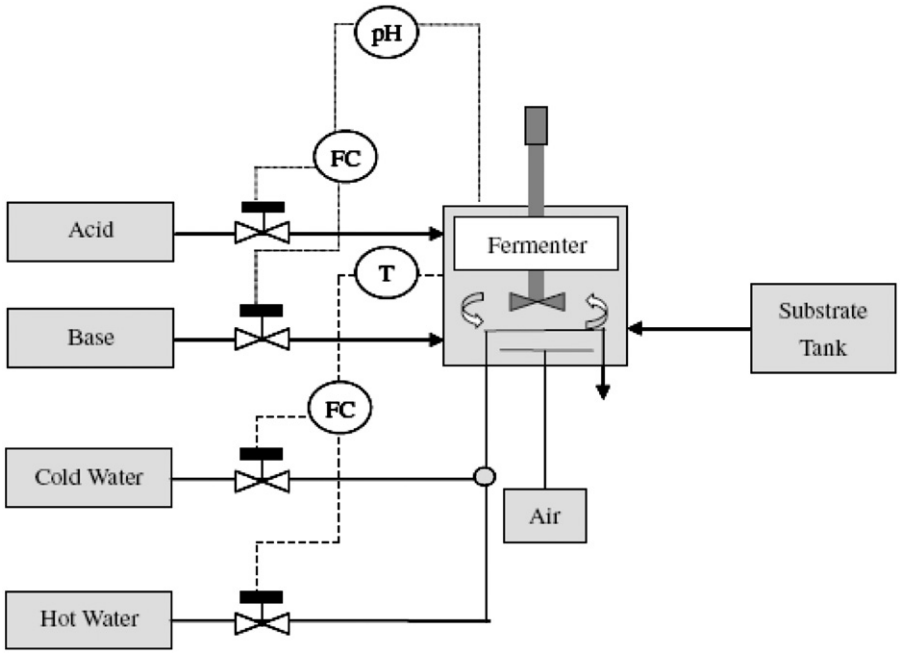

Fig. 5. Penicillin fermentation process.

6. Conclusions

In this article, the recursive KPCA (RKPCA) algorithm is proposed to adapt for time-varying systems. Compared to KPCA, the model build by RKPCA can be updated using the recursive eigenvalues and eigenvectors in the kernel space. The differences between the proposed method and the existing work in this article are as follows:

(1) In the existing work, after the new sample was obtained, the whole model was re-built using the existing method. The model of history data was given up. In this article, the model of history data is adopted to build new model.
(2) In the existing recursive methods, only linear methods were proposed. Because the kernel function is unknown, it is difficult to describe the nonlinear dynamic data structure using kernel trick. In this article, new nonlinear modeling method is proposed based on a new singular value decomposition (SVD) technique.

The proposed method is applied to the fault detection of the continuous annealing process and the penicillin fermentation process. The examples demonstrate that the proposed method can efficiently capture the time-varying and nonlinear relationship in process variables. Thus false alarms and missing detection rate are reduced.

Acknowledgements

The work is supported by China's National 973 program (2009CB320600), the NSF (60974057 and 61020106003) and the Foundation of Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, PR China.

Appendix A

The *SPE* statistic is defined as

$$\begin{aligned} SPE &= \|\boldsymbol{\Phi}(\mathbf{x}_{new})-\hat{\boldsymbol{\Phi}}(\mathbf{x}_{new})\|^2 \\ &= \boldsymbol{\Phi}(\mathbf{x}_{new})\boldsymbol{\Phi}(\mathbf{x}_{new})^T-2\boldsymbol{\Phi}(\mathbf{x}_{new})\hat{\boldsymbol{\Phi}}(\mathbf{x}_{new})^T+\hat{\boldsymbol{\Phi}}(\mathbf{x}_{new})\hat{\boldsymbol{\Phi}}(\mathbf{x}_{new})^T \\ &= k(\mathbf{x}_{new},\mathbf{x}_{new})-2\boldsymbol{\Phi}(\mathbf{x}_{new})\mathbf{P}\mathbf{t}_{new}^T+\mathbf{t}_{new}\mathbf{P}^T\mathbf{P}\mathbf{t}_{new}^T \\ &= 1-2\mathbf{k}_{new}\mathbf{A}\mathbf{t}_{new}^T+\mathbf{t}_{new}\mathbf{A}^T\mathbf{K}\mathbf{A}\mathbf{t}_{new}^T \end{aligned} \tag{6}$$

The T^2 statistic follows an F-distribution and the confidence limit, T_β^2, is given by

$$T_\beta^2 = \frac{r(N^2-1)}{N(N-r)}F_{r,N-r,\beta} \tag{7}$$

where β is the confidence level. Based on the work in Wang et al. (2003), the SPE statistic can be approximated by a central χ^2-distribution. The confidence limit for SPE, SPE_β, can be approximated by

$$\mathrm{SPE}_\beta = g\chi^2(h) \quad g=\frac{\rho^2}{2\mu} \quad h=\frac{2\mu^2}{\rho^2} \tag{8}$$

where μ and ρ^2 are the mean and the variance of the SPE statistic, respectively.

Appendix B

Let $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$ be the sample matrix. To obtain dynamical relation, $\boldsymbol{X}\otimes\boldsymbol{\beta}=[\boldsymbol{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_0,\boldsymbol{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_1,\ldots,\boldsymbol{X}[(1{:}m,1{:}N)]\otimes\boldsymbol{\beta}_{k-1}]_{m\times k}$ is the dynamic sample matrix, where $\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_0 & \cdots & \boldsymbol{\beta}_{k-1}\end{bmatrix}$, $\boldsymbol{\beta}_0=\begin{bmatrix}0 & 0 & \ldots & 1\end{bmatrix}_{N\times 1}^T$, $\boldsymbol{\beta}_1=\begin{bmatrix}0 & \ldots & 1 & 0\end{bmatrix}_{N\times 1}^T$

$$\boldsymbol{\beta}_{k-1}=\left[\begin{array}{ccccccc}0 & \ldots & 0 & \underbrace{1} & 0 & \ldots & 0\\ & & & k & & & \end{array}\right]_{N\times 1}^T,$$

$\boldsymbol{\beta}_{N-1}=\begin{bmatrix}1 & 0 & \ldots & 0\end{bmatrix}_{N\times 1}^T$ whose singular value decomposition satisfies $\mathbf{X}\otimes\boldsymbol{\beta}=\mathbf{S}\boldsymbol{\Lambda}\mathbf{D}^T$. Let $\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}=[\mathbf{X}[(1:m,1:N-1)]\otimes\boldsymbol{\beta}_0,\ldots,\mathbf{X}[(1:m,1:N-1)]\otimes\boldsymbol{\beta}_{k-1}]$ be the intermediate matrix, whose singular value decomposition satisfies $\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}=\tilde{\mathbf{S}}\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{D}}^T$. $\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}$ can be approximated by

$$\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}\approx(\tilde{\mathbf{X}}\otimes\boldsymbol{\beta})_m=\tilde{\mathbf{S}}_m\tilde{\boldsymbol{\Lambda}}_m\tilde{\mathbf{D}}_m^T$$

where $\tilde{\mathbf{S}}_m$ and $\tilde{\mathbf{D}}_m$ represent the front m columns of $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{D}}$, respectively, $\tilde{\boldsymbol{\Lambda}}_m=diag(\sqrt{\lambda_1},\sqrt{\lambda_2},\cdots,\sqrt{\lambda_m})$, $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_m>0$ represent the front m eigenvalues of $(\tilde{\mathbf{X}}\otimes\boldsymbol{\beta})^T(\tilde{\mathbf{X}}\otimes\boldsymbol{\beta})$. So

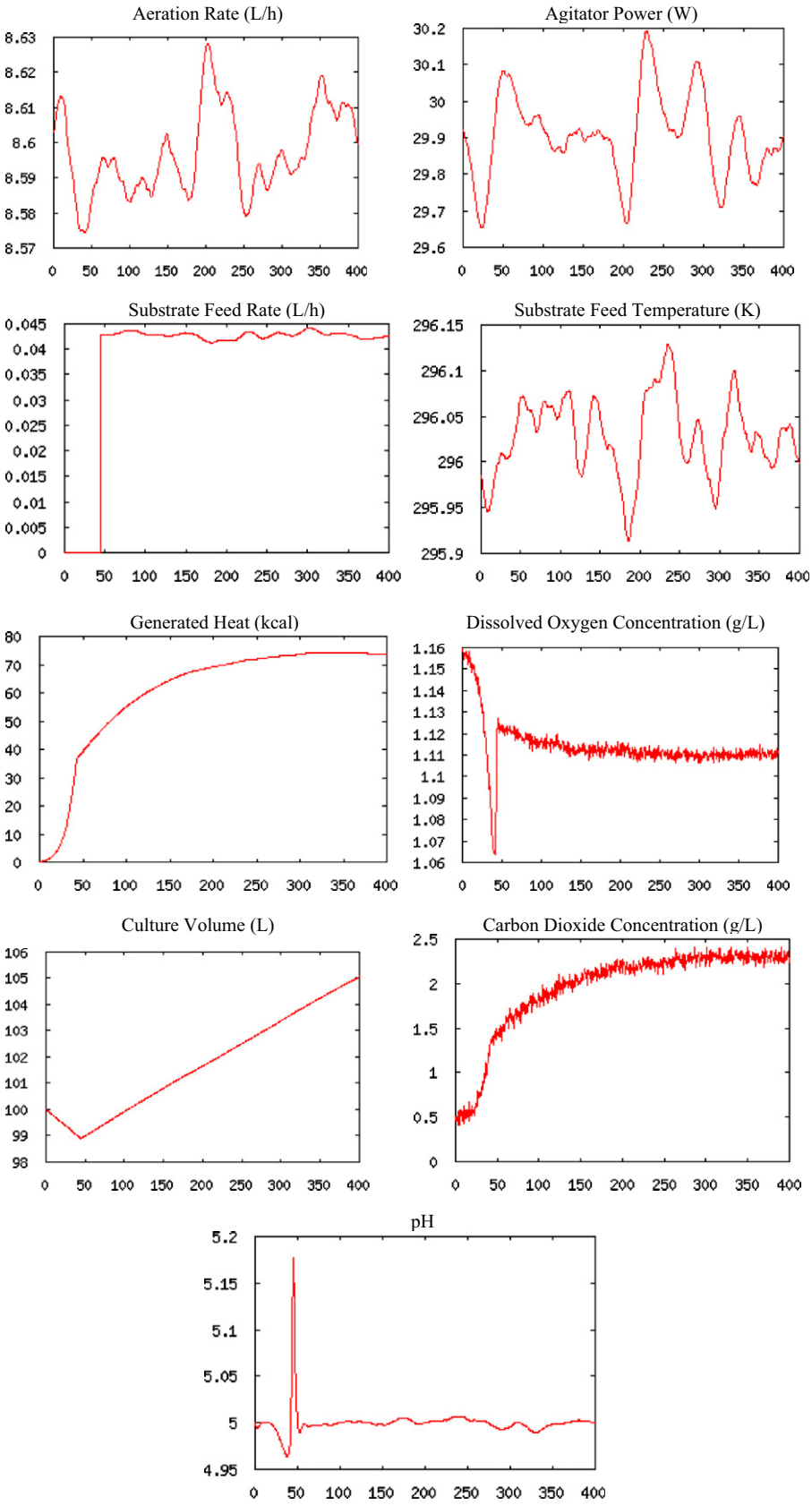

Fig. 6. Trajectories of nine variables from a nominal batch run.

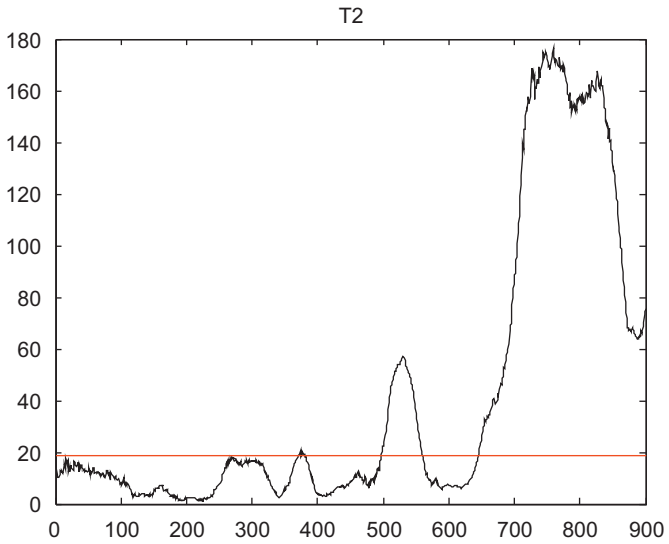

Fig. 7. KPCA monitoring results of the Penicillin fermentation process.

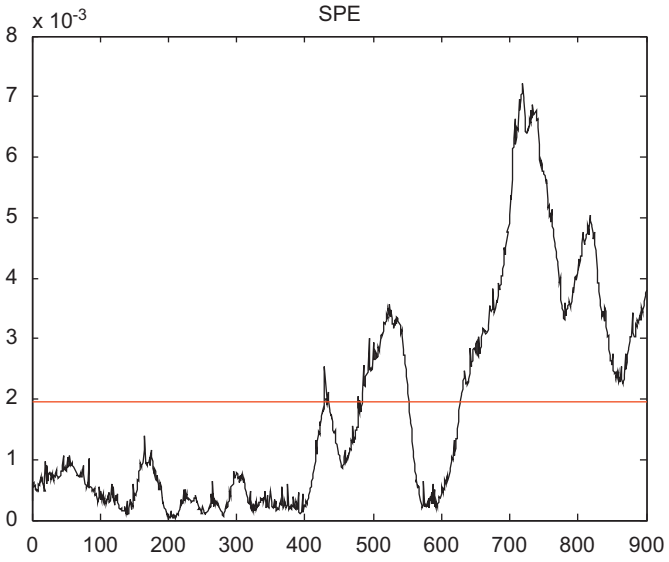

Fig. 8. KPCA monitoring results of the Penicillin fermentation process.

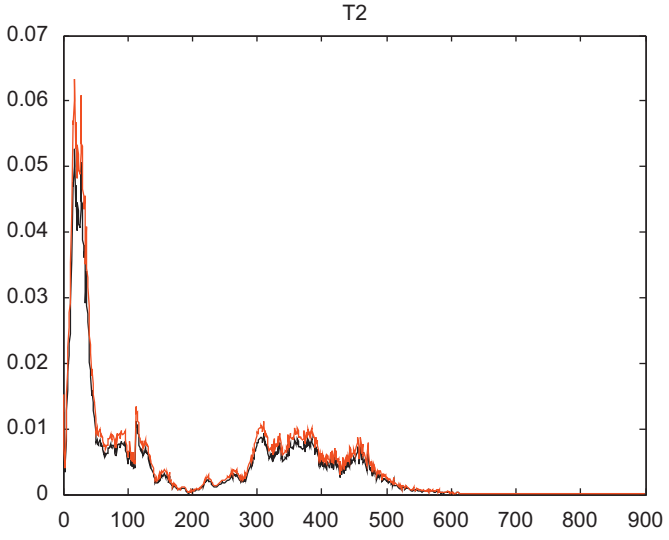

Fig. 9. RKPCA monitoring results of the Penicillin fermentation process.

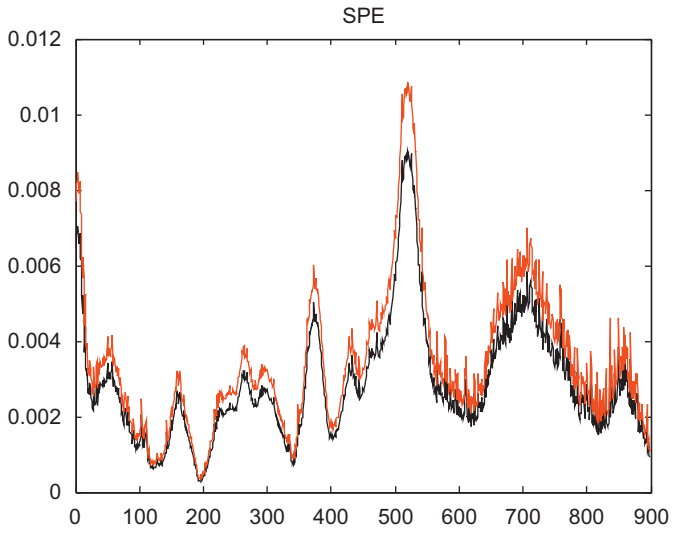

Fig. 10. RKPCA monitoring results of the Penicillin fermentation process.

$\mathbf{X}$ can be represented by

$$\mathbf{X}\otimes\boldsymbol{\beta}=\left[\mathbf{X}[(1:m,1)]\otimes\boldsymbol{\beta}\quad\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}\right]$$

$$=\left[\mathbf{X}[(1:m,1)]\otimes\boldsymbol{\beta}\quad\tilde{\mathbf{S}}_m\right]\begin{bmatrix}1 & \mathbf{0}_m^T\\ \mathbf{0}_m & \tilde{\boldsymbol{\Lambda}}_m\end{bmatrix}\begin{bmatrix}1 & \mathbf{0}_m^T\\ \mathbf{0}_{k-1} & \tilde{\mathbf{D}}_m\end{bmatrix}^T \tag{9}$$

where $\mathbf{0}_m$ is a m-dimensional vector of zeros. Set $\mathbf{V}=\begin{bmatrix}1 & \mathbf{0}_m^T\\ \mathbf{0}_m & \tilde{\boldsymbol{\Lambda}}_m\end{bmatrix}$, which is block symmetric and whose singular value decomposition, is computed:

$$\mathbf{V}=\mathbf{S}'\boldsymbol{\Lambda}'\mathbf{D}'^T \tag{10}$$

The decomposition matrixes are block symmetric such that the decomposition becomes simple. Substituting Eq. (10) into Eq. (9) gives rise to

$$\mathbf{X}\otimes\boldsymbol{\beta}=\left[\mathbf{X}[(1:m,1)]\otimes\boldsymbol{\beta}\quad\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}\right]$$

$$=\left[\mathbf{X}[(1:m,1)]\otimes\boldsymbol{\beta}\quad\tilde{\mathbf{S}}_m\right]\mathbf{S}'\boldsymbol{\Lambda}'\mathbf{D}'^T\begin{bmatrix}1 & \mathbf{0}_m^T\\ 0_{k-1} & \tilde{\mathbf{D}}_m\end{bmatrix}^T=S\boldsymbol{\Lambda}\mathbf{D}^T \tag{11}$$

Then $\tilde{\mathbf{S}}_m$ can be computed according to Eq. (12)

$$\left[\mathbf{X}[(1:m,1)]\otimes\boldsymbol{\beta}\quad\tilde{\mathbf{S}}_m\right]=\mathbf{S}(\mathbf{S}')^{-1} \tag{12}$$

$\mathbf{x}_{new}$ is a new sample, the singular value decomposition of the updated sample matrix $\left[\left[\tilde{\mathbf{X}}\quad\mathbf{x}_{new}\right]\otimes\boldsymbol{\beta}\right]$ can be represented by

$$\left[\left[\tilde{\mathbf{X}}\quad\mathbf{x}_{new}\right]\otimes\boldsymbol{\beta}\right]=\left[\tilde{\mathbf{S}}_m\quad\mathbf{x}_{new}\otimes\boldsymbol{\beta}\right]\begin{bmatrix}\tilde{\boldsymbol{\Lambda}}_m & 0_m\\ 0_m^T & 1\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{D}}_m & 0_{k-1}\\ 0_m^T & 1\end{bmatrix}^T \tag{13}$$

Set $\hat{\mathbf{V}}=\begin{bmatrix}\tilde{\boldsymbol{\Lambda}}_m & 0_m\\ 0_m^T & 1\end{bmatrix}$, whose singular value decomposition is computed:

$$\hat{\mathbf{V}}=\hat{\mathbf{S}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{D}}^T \tag{14}$$

Substituting Eq. (14) into Eq. (13) gives rise to

$$\left[\tilde{\mathbf{X}}\otimes\boldsymbol{\beta}\quad\mathbf{x}_{new}\otimes\boldsymbol{\beta}\right]=\left[\tilde{\mathbf{S}}_m\quad\mathbf{x}_{new}\otimes\boldsymbol{\beta}\right]\hat{\mathbf{S}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{D}}^T\begin{bmatrix}\tilde{\mathbf{D}}_m & 0_{k-1}\\ 0_m^T & 1\end{bmatrix}^T$$

$$=\mathbf{S}''\boldsymbol{\Lambda}''\mathbf{D}''^T \tag{15}$$

where

$$\mathbf{S}'' = \begin{bmatrix} \tilde{\mathbf{S}}_m & \mathbf{x}_{new} \otimes \boldsymbol{\beta} \end{bmatrix} \hat{\mathbf{S}} \tag{16}$$

and

$$\mathbf{D}'' = \hat{\mathbf{D}}^T \begin{bmatrix} \tilde{\mathbf{D}}_\Phi & 0_{k-1} \\ 0_m^T & 1 \end{bmatrix}^T \tag{17}$$

and $\mathbf{D}''$ can be determined by Eq. (8). Set $\mathbf{X}_{new} = \begin{bmatrix} \tilde{\mathbf{X}} & \mathbf{x}_{new} \end{bmatrix} \otimes \boldsymbol{\beta}$, $\mathbf{S}=\mathbf{S}''$, $\mathbf{D}=\mathbf{D}''$, using the above Eqs. (9), (11)–(13) and (15), the updating of eigenvalues and eigenvectors can be computed for the new sample.

Appendix C

When recursive computing process is extended to the feature space ($\mathbf{F}$), a new recursive KPCA method is proposed here to update the eigenvalue decomposition of the covariance matrix $\mathbf{C}^F$. λ_i and $\mathbf{p}_i$ represent the ith eigenvalue and eigenvector of $\mathbf{C}^F$. The centered Gram matrix is

$$\mathbf{K} = \overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta})^T \overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta}) \tag{18}$$

$\mathbf{v}_i$ represents the ith eigenvector of K. Thus

$$\mathbf{p}_i = \overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta})\mathbf{v}_i \tag{19}$$

$$\begin{aligned} \mathbf{P}_m &= [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m] = \overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta})[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m] = \overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta})\mathbf{V}_m \\ &= \Phi(\mathbf{X} \otimes \boldsymbol{\beta})\mathbf{A}_m \end{aligned} \tag{20}$$

where $\mathbf{A}_m = (\mathbf{I} - (1/k)\boldsymbol{E}_k)\boldsymbol{V}_m$.

Set $\boldsymbol{\Phi}(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta}) = [\boldsymbol{\Phi}(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta}_0), \ldots, \boldsymbol{\Phi}(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta}_{k-1})]$, according to Eqs. (8) and (20), $\overline{\Phi}(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta})$ can be approximated by

$$\overline{\Phi}(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta}) = \tilde{\mathbf{S}}_\Phi \tilde{\boldsymbol{\Lambda}}_\Phi \tilde{\mathbf{D}}^T{}_\Phi \tag{21}$$

where $\tilde{\mathbf{S}}_\Phi = \Phi(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta})\tilde{\mathbf{A}}_\Phi$, $\tilde{\boldsymbol{\Lambda}}_\Phi$ denote the m dominant eigenvalues of the kernel matrix. According to Eqs. (9) and (21), we get

$$\overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta}) = \begin{bmatrix} (\Phi(\mathbf{x}_1) - \tilde{\mathbf{C}}) & \tilde{\mathbf{S}}_\Phi \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}_m^T \\ \mathbf{0}_m & \tilde{\boldsymbol{\Lambda}}_\Phi \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}_m^T \\ \mathbf{0}_{N-1} & \tilde{\mathbf{D}}_\Phi \end{bmatrix}^T \tag{22}$$

where $\tilde{\mathbf{C}} = (1/k-1)\Phi(\tilde{\mathbf{X}} \otimes \boldsymbol{\beta})\mathbf{1}_{k-1}$. Set $\mathbf{V}_\Phi = \begin{bmatrix} 1 & \mathbf{0}_m^T \\ \mathbf{0}_m & \tilde{\boldsymbol{\Lambda}}_\Phi \end{bmatrix}$, whose singular value decomposition is computed:

$$\mathbf{V}_\Phi = \mathbf{S}'_\Phi \boldsymbol{\Lambda}'_\Phi \mathbf{D}'_\Phi{}^T \tag{23}$$

Substituting Eq. (23) into Eq. (22) gives rise to

$$\begin{aligned} \overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta}) &= \Phi(\mathbf{X} \\ &\otimes \boldsymbol{\beta}) \begin{bmatrix} 1 & \mathbf{0}_m^T \\ -\frac{1}{k-1}\mathbf{1}_{k-1} & \tilde{\mathbf{A}}_\Phi \end{bmatrix} \mathbf{S}'_\Phi \boldsymbol{\Lambda}'_\Phi \mathbf{D}'^T_\Phi \begin{bmatrix} 1 & \mathbf{0}_m^T \\ 0_{k-1} & \tilde{\mathbf{D}}_\Phi \end{bmatrix}^T \end{aligned} \tag{24}$$

sAccording to Eqs. (20) and (21), $\overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta})$ can be represented by

$$\overline{\Phi}(\mathbf{X} \otimes \boldsymbol{\beta}) = \mathbf{S}_\Phi \mathbf{V}_\Phi \mathbf{D}_\Phi^T = \Phi(\mathbf{X} \otimes \boldsymbol{\beta})\mathbf{A}\boldsymbol{\Lambda}'_\Phi \mathbf{D}_\Phi^T \begin{bmatrix} 1 & \mathbf{0}_m^T \\ 0_{k-1} & \tilde{\mathbf{D}}_\Phi \end{bmatrix}^T \tag{25}$$

where $\mathbf{S}_\Phi = \Phi(\mathbf{X} \otimes \boldsymbol{\beta})\mathbf{A}$, $\mathbf{D}_\Phi = \begin{bmatrix} 1 & \mathbf{0}_m^T \\ 0_{k-1} & \tilde{\mathbf{D}}_\Phi \end{bmatrix} \mathbf{D}'_\Phi$, because Eq. (24) equals to Eq. (25), we get

$$\mathbf{A} = \begin{bmatrix} 1 & \mathbf{0}_m^T \\ -\frac{1}{k-1}\mathbf{1}_{k-1} & \tilde{\mathbf{A}}_m \end{bmatrix} \mathbf{S}'_\Phi \tag{26}$$

Then $\tilde{\mathbf{A}}_\Phi$ can be computed according to Eq. (26):

$$\begin{bmatrix} 1 & \mathbf{0}_m^T \\ -\frac{1}{k-1}\mathbf{1}_{k-1} & \tilde{\mathbf{A}}_\Phi \end{bmatrix} = \mathbf{A}(\mathbf{S}'_\Phi)^{-1} \tag{27}$$

Assume $\Phi(\begin{bmatrix} \tilde{\mathbf{X}} & \mathbf{x}_{new} \end{bmatrix} \otimes \boldsymbol{\beta})$ is the new sample transformed into the feature space, $\Phi(\mathbf{X}_{new}) = \Phi(\begin{bmatrix} \tilde{\mathbf{X}} & \mathbf{x}_{new} \end{bmatrix} \otimes \boldsymbol{\beta})$ is the updated sample matrix. Then the mean vector and the covariance matrix of $\boldsymbol{\Phi}(\mathbf{X}_{new})$ can be computed:

$$\overline{\Phi}(\mathbf{X}_{new}) = \frac{1}{k-1}\left[\Phi(\begin{bmatrix} \tilde{\mathbf{X}} & \mathbf{x}_{new} \end{bmatrix} \otimes \boldsymbol{\beta}) - \tilde{\mathbf{C}}_{new}\right]\left[\Phi(\begin{bmatrix} \tilde{\mathbf{X}} & \mathbf{x}_{new} \end{bmatrix} \otimes \boldsymbol{\beta}) - \tilde{\mathbf{C}}_{new}\right]^T \tag{28}$$

According to Eqs. (13), (21) and (28), we get

$$\overline{\Phi}(\mathbf{X}_{new}) = \begin{bmatrix} \tilde{\mathbf{S}}_\Phi & \Phi(\mathbf{x}_{new} \otimes \boldsymbol{\beta}) - \tilde{\mathbf{C}}_{new} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{V}}_\Phi & \mathbf{0}_m \\ \mathbf{0}_m^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{D}}_\Phi & \mathbf{0}_{k-1} \\ \mathbf{0}_m^T & 1 \end{bmatrix}^T \tag{29}$$

set $\hat{\mathbf{V}}_\Phi = \begin{bmatrix} \tilde{\mathbf{V}}_\Phi & \mathbf{0}_m \\ \mathbf{0}_m^T & 1 \end{bmatrix}$, whose singular value decomposition is computed:

$$\hat{\mathbf{V}}_\Phi = \hat{\mathbf{S}}_\Phi \hat{\mathbf{V}}_\Phi \hat{\mathbf{D}}_\Phi^T \tag{30}$$

Substituting Eq. (30) into Eq. (29), Then eigenvectors $\mathbf{P}_{\overline{\Phi}(\mathbf{X}_{new})}$ of $\overline{\Phi}(\mathbf{X}_{new})$ is obtained. The KPCA score vector $\mathbf{t}$, for a new sample $\mathbf{x}_{new}$ is given by

$$\mathbf{t} = (\mathbf{P}_{\overline{\Phi}(\mathbf{X}_{new})})^T \left[\Phi(\begin{bmatrix} \tilde{\mathbf{X}} & \mathbf{x}_{new} \end{bmatrix} \otimes \boldsymbol{\beta}) - \tilde{\mathbf{C}}_{new}\right] \tag{31}$$

References

AlGhazzawi, A., Lennox, B., 2008. Monitoring a complex refining process using multivariate statistics. Control Eng. Pract. 16 (3), 294–307.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Auto. Con. 19, 716–723.

Chiang, L.H., Russell, F.L., Braatz, R.D., 2001. Fault detection and diagnosis in industrial systems. Springer, London.

Cattell, R.B., 1966. The scree test for the number of factors. Multivariate Behav. Res 1, 245.

Cheng, C.Y., Hsu, C.C., Chen, M.C., 2010. Adaptive Kernel Principal Component Analysis (KPCA) for Monitoring Small Disturbances of Nonlinear Processes. Ind. Eng. Chem. Res. 49, 2254–2262.

Dong, D., McAvoy, T.J., 1996. Nonlinear principal component analysis-based on principal curves and neural networks. Comput. Chem. Eng. 20 (1), 65–78.

Elshenawy, L.M., Yin, S., Naik, A.S., Ding, S.X., 2010. Efficient recursive principal component analysis algorithms for process monitoring. Ind. Eng. Chem. Res. 49, 252–259.

Ge, Z., Song, Z., 2007. Process monitoring based on independent component analysis–principal component analysis (ICA–PCA) and similarity factors. Ind. Eng. Chem. Res. 46 (7), 2054–2063.

Gallagher, V.B., Wise, R.M., Butler, S.W., White, D.D., Barna, G.G., Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process; improving robustness through model updating, in: Proc. of ADCHEM 97, Banff, Canada, 9–11 June 1997, pp. 78-83.

Ge, Z., Yang, C., Song, Z., 2009. Improved kernel PCA-based monitoring approach for nonlinear processes. Chem. Eng. Sci. 64 (9), 2245–2255.

Gajardo, P., Harmand, J., Ramírez, C.H., Rapaport, A., 2011. Minimal time bioremediation of natural water resources. Automatica 47 (8), 1764–1769.

Hall, P., Marshall, D., Martin, R., 2000. Merging and splitting eigenspace models. IEEE Trans. Pattern. Anal. Mach. Intell. 22, 1042–1049.

Jia, F., Martin, E.B., Morris, A.J., 2000. Nonlinear principal components analysis with application to process fault detection. Int. J. Syst. Sci. 31 (11), 1473–1487.

Johnson, A., 1987. The control of fed-batch fermentation processes—A survey Original Research Article. Automatica 23 (6), 691–705.

Jeng, J.–C., Li, C.–C., Huang, H.-P., 2007. Fault detection and isolation for dynamic processes using recursive principal component analysis (PCA) based on filtering of signals. Asia-Pacific J. Chem. Eng. 2 (6), 501–509.

Kruger, U., Chen, Q., Sandoz, D.J., McFarlane, R.C., 2001. Extended PLS approach for enhanced condition monitoring of industrial processes. Aiche J. 47 (9), 2076–2091.

Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., Ohno, H., 2003. Monitoring independent components for fault detection. AIChE J. 49, 969–976.

Kramer, M.A., 1991. Nonlinear principal component analysis using autoassociative neural networks. Aiche J. 37 (3), 233–243.

Kruger, U., Zhang, J., Xie, L., 2007. Principal manifolds for data visualization and dimension reduction. vol. 58 of Lecture notes in computational science and engineering. Springer Verlag, Berlin-Heidelberg-New York, Ch. Developments and applications of nonlinear principal component analysis—A review, pp. 1-44.

Lee, J.M., Qin, S.J., Lee, I.B., 2006. Fault detection and diagnosis based on modified independent component analysis. Aiche J. 52 (10), 3501–3514.

Liu, X., Kruger, U., Littler, T., Xie, L., Wang, S.–Q., 2009. Moving window kernel PCA for adaptive monitoring of nonlinear processes. Chemometrics and Intelligent Laboratory Systems 96 (2), 132–143.

Morud, T.E., 1996. Multivariate statistical process control; example from the chemical process industry. J. Chemometr. 10 (5–6), 669–675.

Malinowski, F.R., 1991. Factor Analysis in Chemistry. Wiley-Inter-science, New York.

Maestri, M.L., Cassanello, M.C., Hororwitz, G.I., 2009. Kernel PCA performance in processes with multiple operation modes. Chemical Product and Process Modeling 4 (5).

Nomikos, P., MacGregor, J.F., 1995. Multivariate spc charts for monitoring batch processes. Technometrics 37 (1), 41–59.

Osten, D.W., 1988. Selection of optimal regression models via cross- validation. J. Chemometr. 2, 39–48.

Scholkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Muller, K.R., Ratsch, G., 1999. Input space versus feature space in kernel-based methods. IEEE Trans. Neural Networks 10 (5), 1000–1016.

Schölkopf, B., Smola, A.J., Müller, K., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10 (5), 1299–1399.

Shrager, R.I., Hendler, R.W., 1982. Titration of individual components in a mixture with resolution of difference spectra, pks, and redox transition. Anal. Chem. 54 (7), 1147–1152.

Tan, S., Mavrovouniotis, M.L., 1995. Reducing data dimensionality through optimizing neural network inputs. Aiche J. 41 (6), 1471–1480.

Tracy, N.D., Young, J.C., Mason, R.L., 1992. Multivariate control charts for individual observations. J. Quality Control 24 (2), 88–95.

Voegtlin, T., 2005. Recursive principal components analysis. Neural Net. 18 (8), 1051–1063.

Wang, X., Kruger, U., Irwin, G.W., 2005. Process monitoring approach using fast moving window PCA. Ind. Eng. Chem. Res. 44 (15), 5691–5702.

Wold, S., 1978. Cross validatory estimation of the number of components in factor and principal component analysis. Technometrics 20, 397–406.

Wang, X., Kruger, U., Lennox, B., 2003. Recursive partial least squares algorithms for monitoring complex industrial processes. Control Eng. Pract. 11 (6), 613–632.

Xu, G., Kailath, T., 1994. Fast estimation of principal eigenspace using Lanczos algorithm. SIAM J. Matrix Anal. Appl 15, 974–994.

Zhang, J., Zhao, S.J., Xu, Y.M., 2006. Performance monitoring of processes with multiple operating modes through multiple pls models. J. Process Control 16 (7), 763–772.

Zhang, Y.W., Qin, S.J., 2007. Fault detection of nonlinear processes using multiway kernel independent analysis. Ind. Eng. Che. Res 46, 7780–7787.

Zhang, Y., Qin, S.J., 2008. Improved nonlinear fault detection technique and statistical analysis. Aiche J. 12, 3207–3220.

Zhang, Y.W., Ma, C., 2011. Fault diagnosis of nonlinear processes using multiscale PCA and multiscale KPLS. Chem. Eng. Sci. 66, 64–72.

Zhang, Y.W., Fan, Y.P., Zhang, P.C., 2010. Combining kernel partial least-squares modeling and iterative learning control for the batch-to-batch optimization of constrained nonlinear processes. Ind. Eng. Chem. Res. 9 (16), 7470–7477.

Zhang, Y.W., Zhou, H., Qin, S.J., Chai, T.Y., 2010. Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. IEEE Trans. Ind. Inform. 1, 3–12.