

# On-line Monitoring of Batch Processes Using Additive Kernel Partial Least Square

Ziang Ma, Huangang Wang and Junwu Zhou

## 1 Introduction

Batch processes are widely used in modern industry like biochemical, foods and medicines [3]. In Batch process, raw materials are added in batches and the whole process can be divided into a number of stages where different products are made. To ensure the safety of process, monitoring methods gain much attention. Those monitoring methods like principal component analysis (PCA) [6], support vector domain description (SVDD) [2] only utilize process variables to train the model and test abnormality. Besides process variables, we can obtain quality variables at the end of batches. Therefore, partial least square (PLS) [7] gains people's attention because it considers both process variables and quality variables.

The PLS finds a series of projection directions of both process variables and quality variables. We hope that projected vectors of process variables into projection directions have high correlation with projected vectors obtained from quality variables. Those projection directions can be found through several times of iterations. After that projected vectors and residual vector can be obtained and both  $T^2$  and SPE statistic can be derived. We can compare those statistics with predefined control line to judge the condition of the process. Additionally, kernel partial least square (KPLS) is introduced if we want to analyze non-linear relationship between process variables and quality variables.

---

Z. Ma · H. Wang (✉)

Department of Automation, Tsinghua University, Beijing, China  
e-mail: hgwang@tsinghua.edu.cn

Z. Ma

e-mail: maza11@126.com

J. Zhou

State Key Laboratory of Process Automation in Mining and Metallurgy, Beijing, China  
e-mail: zhou\_jw@bgrimm.com

© Springer Nature Singapore Pte Ltd. 2018

Z. Deng (ed.), *Proceedings of 2017 Chinese Intelligent Automation Conference*,  
Lecture Notes in Electrical Engineering 458,  
[https://doi.org/10.1007/978-981-10-6445-6\\_30](https://doi.org/10.1007/978-981-10-6445-6_30)

In on-line monitoring, only data until current time point are available, which is a kind of missing data problem [5]. In order to deal with missing data, researches are made to estimate statistics when data is incomplete. One handling method is filling those missing data with predefined data, average of training data or current obtained data for instance. Another method makes use of the characteristics of monitoring method. As for PLS, we can estimate projected vectors based on the thought of least square. Then we can calculate  $T^2$  and SPE statistics. However, if nonlinear kernel like Gaussian kernel is chosen as kernel in KPLS, we cannot use the thought of least square to estimate projected vectors because the value of kernel cannot be decomposed into sum of kernel of every time slice.

The additive kernel [4, 10] is a special kind of kernel. It considers nonlinear relationships among variables in every time slice and the value of generalized additive kernel can be decomposed into sum of kernel of every time slice. Based on characteristics mentioned above, we can continue to use the thought of least square to estimate projected vectors without filling missing data. In this paper, we will introduce additive kernel into KPLS [9] and name it AKPLS. Then we can also estimate projected vectors without filling those missing data.

## 2 Kernel Partial Least Square Based Batch Processes Monitoring

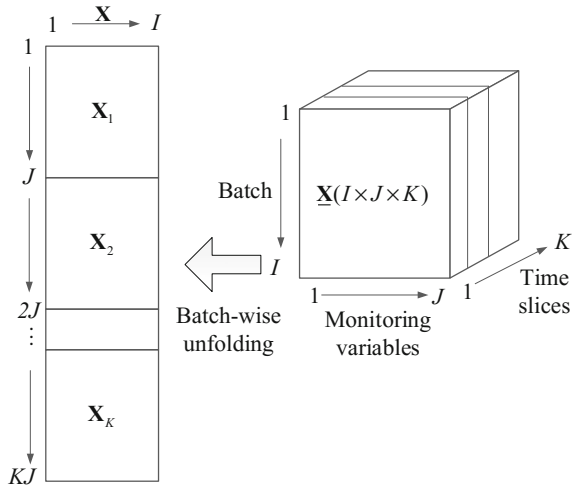
Sample data in batch processes are usually expressed as the three-dimensional matrix  $\underline{\mathbf{X}}(I \times J \times K)$ , where  $I$  denotes the number of batches,  $J$  denotes the number of variables at each time point,  $K$  denotes the number of time slices in a batch. Before using monitoring methods, the data preprocessing for batch processes is needed [8]. In data preprocessing, we unfold the three-dimensional matrix  $\underline{\mathbf{X}}$  and get a two-dimensional matrix  $\mathbf{X}(I \times JK)$ . Figure 1 represents the process of batch-wise unfolding method. Then we centralize and normalize different variables and get process variable matrix  $\mathbf{X}$  and quality variable matrix  $\mathbf{Y}$ .

PLS considers linear relationships between  $\mathbf{X}$  and  $\mathbf{Y}$ , like (1) where  $t_i$  denotes projection score,  $\mathbf{p}_i$  and  $\mathbf{q}_i$  are loading vectors,  $\mathbf{E}$  and  $\mathbf{F}$  are residuals.

$$\mathbf{X} = \sum_{i=1}^n t_i \mathbf{p}_i^T + \mathbf{E}, \mathbf{Y} = \sum_{i=1}^n t_i \mathbf{q}_i^T + \mathbf{F} \quad (1)$$

For each preprocessed test sample  $\mathbf{x} \in R^{m \times 1}$ , we can derive  $n$  projected scores and residual vector. The iteration procedures to derive statistics from complete test sample using PLS are summarized as follow:

**Fig. 1** Batch-wise unfolding process



1.  $\mathbf{x}_1 = \mathbf{x}, i = 1.$
2. When  $i \leq n,$  do the following steps 3.
3.  $t_i = \mathbf{x}_i^T \mathbf{w}_i, \mathbf{x}_{i+1} = \mathbf{x}_i - t_i \mathbf{p}_i, i = i + 1.$
4. Set  $\mathbf{t} = [t_1 \ t_2 \ \dots \ t_n]^T$  as projected vector,  $\mathbf{x}_{n+1}$  as residual vector.
5. Set  $\mathbf{t}^T \mathbf{\Lambda}^{-1} \mathbf{t}$  as  $T^2$  statistic ( $\mathbf{\Lambda}$  denotes diagonal matrix of the variances of the projected vectors). Set SPE statistic as square of  $\mathbf{x}_{n+1}.$

The control limit of  $T^2$  can be estimated by F-distribution and the control limit of SPE can be estimated as chi-squared distribution, like (2), where  $N$  denotes the number of training samples,  $n$  denotes the number of projection directions,  $\alpha$  denotes the significance level.  $m$  and  $v$  denote the mean and variance of SPE calculated from training data.

$$T^2 \sim \frac{n(N^2 - 1)}{N(N - n)} F_{n, N-n, \alpha} \tag{2}$$

$$SPE \sim g \chi_{h, \alpha}^2, g = v/(2m), h = 2m^2/v$$

In batch process, there exists non-linear relationship among process variables and quality variables. Therefore, we introduce kernel to make it KPLS by replacing  $\mathbf{X}$  with  $\Phi(\mathbf{X})$  using (3).

$$\Phi(\mathbf{X}) = [\Phi(\mathbf{x}_1) \ \Phi(\mathbf{x}_2) \ \dots \ \Phi(\mathbf{x}_N)]^T \tag{3}$$

$$Ker(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

In batch process on-line monitoring, data after current time point are unavailable. If we choose traditional nonlinear kernel like Gaussian kernel and polynomial kernel, those missing data should be filled before we calculate monitoring statistics.

Inappropriate padding may affect the monitoring result. In the next section, another special kernel called additive kernel is introduced to solve this problem.

### 3 Batch Process Monitoring Based on Additive Kernel Partial Least Square (AKPLS)

In this section, we introduce a kind of kernel called additive kernel (AK) and the training and monitoring procedure of AKPLS.

#### 3.1 Additive Kernel (AK)

Additive Kernel (AK) is a special kind of kernel which is defined as (4), where  $K$  denotes  $K$  different time slices.

$$\Phi(\mathbf{x}_i) = [\phi(\mathbf{x}_{i,1})^T \quad \phi(\mathbf{x}_{i,2})^T \quad \dots \quad \phi(\mathbf{x}_{i,K})^T]^T \quad (4)$$

Data from different time slices are mapped independently in  $\Phi(\mathbf{x}_i)$ . The value of the whole kernel can be decomposed into the sum of  $K$  kernels. We can select different kind of kernels at different time slices to analyze non-linear relationship among variables. If kernel at every time slice is Gaussian kernel, it is called additive Gaussian kernel, like (5).

$$\begin{aligned} Ker(\mathbf{x}_i, \mathbf{x}_j) &= \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) = \sum_{k=1}^K \exp\left(-\|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|^2 / r\right) \\ \phi(\mathbf{x}_{i,k})^T \phi(\mathbf{x}_{j,k}) &= \exp\left(-\|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|^2 / r\right) \end{aligned} \quad (5)$$

#### 3.2 Training Process of AKPLS

Additive Kernel Partial Least Square(AKPLS) is a special case of KPLS where  $\Phi(\mathbf{x}_i)$  satisfied with (4) can be chosen as kernel. Then we can derive training process of AKPLS, summarized as Algorithm 1.

**Algorithm 1: Training part of AKPLS**

Input:  $\mathbf{X}$  as process variables matrix,  $\mathbf{Y}$  as quality variables matrix,  $n$  as iteration times.

1. Set  $\Phi(\mathbf{X})_1 = \Phi(\mathbf{X})(\mathbf{I} - \frac{1}{h}\mathbf{1}\mathbf{1})$ , where  $\mathbf{I}$  denotes unit matrix,  $\mathbf{1}\mathbf{1}$  denotes matrix with all elements equals to 1,  $h$  is the dimension of the matrix, the definition of  $\Phi(\mathbf{X})$  is (1.3). Set  $\mathbf{Y}_1 = \mathbf{Y}$  and  $i = 1$ .
2. If  $i \leq n$  do the following 3, 4, 5, 6, else do 7.
3. Set  $\mathbf{t}_i$  as the eigenvector of maximum eigenvalue of  $\Phi(\mathbf{X})_i \Phi(\mathbf{X})_i^T \mathbf{Y}_i \mathbf{Y}_i^T$ .
4. Calculate  $\mathbf{c}_i, \mathbf{u}_i, \mathbf{w}_i, \mathbf{t}_i$  using  $\mathbf{c}_i = \mathbf{Y}_i^T \mathbf{t}_i, \mathbf{c}_i = \mathbf{c}_i / \|\mathbf{c}_i\|, \mathbf{u}_i = \mathbf{Y}_i \mathbf{c}_i, \mathbf{w}_i = \Phi(\mathbf{X})_i^T \mathbf{u}_i, \mathbf{w}_i = \mathbf{w}_i / \|\mathbf{w}_i\|, \mathbf{t}_i = \Phi(\mathbf{X})_i \mathbf{w}_i$ .
5. Use  $\mathbf{p}_i = \frac{\Phi(\mathbf{X})_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}, \mathbf{q}_i = \frac{\mathbf{Y}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}, \Phi(\mathbf{X})_{i+1} = \Phi(\mathbf{X})_i - \mathbf{t}_i \mathbf{p}_i^T, \mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{t}_i \mathbf{q}_i^T$  to calculate  $\mathbf{p}_i, \Phi(\mathbf{X})_{i+1}, \mathbf{Y}_{i+1}$ .
6. Set  $i = i + 1$ .
7. Get  $\mathbf{w}_i, \mathbf{p}_i$  ( $i = 1, 2, \dots, n$ ) after  $n$  iterations.
8. Using  $\Phi(\mathbf{X})_{n+1}, \mathbf{t}_i$  ( $i = 1, 2, \dots, n$ ) to calculate  $SPE_k, \overline{SPE}_k, T^2$  statistics of training data and set control limits of those statistics.

Output:  $\mathbf{w}_i, \mathbf{p}_i$  ( $i = 1, 2, \dots, n$ ). Control limit of  $SPE_k, \overline{SPE}_k, T^2$ .

In Algorithm 1, statistic  $T^2$  is defined as  $\mathbf{t}^T \mathbf{\Lambda}^{-1} \mathbf{t}$  in which  $\mathbf{\Lambda}$  denotes diagonal matrix of the variances of the scores associated with  $n$  projected directions.  $SPE_k$  denotes sum of the square of the residual vector corresponding to time point  $k$ .  $\overline{SPE}_k = \sum_{i=1}^k SPE_i / k$ . The control limits of those statistics are shown in (2).

**3.3 On-line Monitoring of AKPLS**

In on-line monitoring, variables after current time slice are unknown. In this part, we will mainly discuss how to estimate monitoring statistics without filling in unknown data in AKPLS.

In AKPLS,  $\Phi(\mathbf{x}_j)$  can be decomposed as (6). At time point  $k, \Phi^{(1,k)}(\mathbf{x}_j)$  is known while  $\Phi^{(k+1,K)}(\mathbf{x}_j)$  is unknown.

$$\begin{aligned}
 \Phi(\mathbf{x}_j) &= [\Phi^{(1,k)}(\mathbf{x}_j)^T \quad \Phi^{(k+1,K)}(\mathbf{x}_j)^T]^T \\
 \Phi^{(1,k)}(\mathbf{x}_j) &= [\phi(\mathbf{x}_{j,1})^T \quad \phi(\mathbf{x}_{j,2})^T \quad \dots \quad \phi(\mathbf{x}_{j,k})^T]^T \\
 \Phi^{(k+1,K)}(\mathbf{x}_j) &= [\phi(\mathbf{x}_{j,k+1})^T \quad \phi(\mathbf{x}_{j,2})^T \quad \dots \quad \phi(\mathbf{x}_{j,K})^T]^T
 \end{aligned}
 \tag{6}$$

In on-line monitoring, we can derive optimization problem (7) to estimate projection vectors of test data  $\mathbf{x}_j$ , with corresponding results shown in (8).  $\hat{t}_i$  shows the estimated projected score.  $\mathbf{w}_i^{(1,k)}$  and  $\mathbf{p}_i^{(1,k)}$  show parts of  $\mathbf{w}_i$  and  $\mathbf{p}_i$

corresponding to times slices from 1 to  $k$ . Then we can derive  $\Phi_{i+1}^{(1,k)}(\mathbf{x}_j)$  with (9). On-line monitoring of AKPLS is shown in Algorithm 2.

$$\min \left\| \Phi_i^{(1,k)}(\mathbf{x}_j)^T - \hat{t}_i \mathbf{w}_i^{(1,k)T} \right\|_2 \tag{7}$$

$$\hat{t}_i = \Phi_i^{(1,k)}(\mathbf{x}_j)^T \mathbf{w}_i^{(1,k)} / (\mathbf{w}_i^{(1,k)T} \mathbf{w}_i^{(1,k)}) \tag{8}$$

$$\Phi_{i+1}^{(1,k)}(\mathbf{x}_j) = \Phi_i^{(1,k)}(\mathbf{x}_j) - \hat{t}_i \mathbf{p}_i^{(1,k)} \tag{9}$$

**Algorithm 2: On-line monitoring of AKPLS**

Input:  $\mathbf{x}^{(1,k)}$  denotes variables of  $\mathbf{x}$  from time slice 1 to  $k$ .  $\mathbf{w}_i, \mathbf{p}_i$  ( $i = 1, 2, \dots, n$ ), control limit of  $SPE_k, \overline{SPE}_k, T^2$ .

1. Set  $i = 1$  and centralize  $\Phi^{(1,k)}(\mathbf{x})$  to get  $\Phi_1^{(1,k)}(\mathbf{x})$ .
2. If  $i \leq n$  do the following steps 3, else do 4.
3. Estimate  $\hat{t}_i$  with (8), estimate  $\Phi_{i+1}^{(1,k)}(\mathbf{x})$  with (9),  $i = i + 1$ .
4. Set  $\mathbf{t} = [\hat{t}_1 \ \hat{t}_2 \ \dots \ \hat{t}_n]^T$ .
5. Calculate monitoring statistics using  $\Phi_{n+1}^{(1,k)}(\mathbf{x})$  and  $\mathbf{t}$ .
6. Compare statistics with control limit and judge whether  $\mathbf{x}$  is normal.

Output: whether the sample is normal or not.

Additive kernel has characteristics that variables in different time slices are independent when mapped into feature space.  $\mathbf{w}_i$  and  $\mathbf{p}_i$  ( $i = 1, 2, \dots, n$ ) are linear combinations of projections in feature space of training data. Therefore,  $\mathbf{w}_i$  and  $\mathbf{p}_i$  can also be decomposed into  $\mathbf{w}_i^{(1,k)}, \mathbf{w}_i^{(k+1,K)}$  and  $\mathbf{p}_i^{(1,k)}, \mathbf{p}_i^{(k+1,K)}$  respectively, similar to (6). However, some nonlinear kernels like Gaussian kernel cannot be decomposed into two parts like (6), so we cannot use methods discussed above to estimate statistics without filling unknown data.

More experiments and analysis about score estimation using the thought of least square are shown in [5], where estimation methods in PLS are discussed.

## 4 Case Study

A fed-batch penicillin fermentation dataset is used to evaluate the performance of methods. The dataset is simulated by a standard simulator Pensim V.2.0 [1].

In my experiments, 110 normal batches and 4 abnormal batches are generated from the simulator. Every batch lasts 400 h and the sampling rate is 30 min. The start time point of faults of 4 abnormal batches are 60, 55, 40, 30 h respectively.

Those 4 faults last until the end of batch. Training dataset includes 100 normal batches. Test dataset includes 10 normal and 4 abnormal batches. Eight variables including Aeration rate, Agitator power, Substrate feed temperature, Culture volume, Carbon dioxide concentration, pH, Bioreactor Temperature, Generated heat are selected as process variables while 3 variables including Substrate concentration, Biomass concentration, Penicillin concentration are quality variables. Every batch includes 6400 process variables and 3 quality variables because we can get quality variables only at the end of every batch.

On-line monitoring starts from the 21st time point until the end of the batch. We choose 21 because a small portion of data is necessary when we estimate test statistics. Faced with missing data problem, PLS and AKPLS uses method discussed in Algorithm 2 to estimate statistics while KPLS fills data after current time with current data before calculating monitoring statistics.  $\overline{SPE}_k$  is selected to judge the test sample because it works better than other statistics.

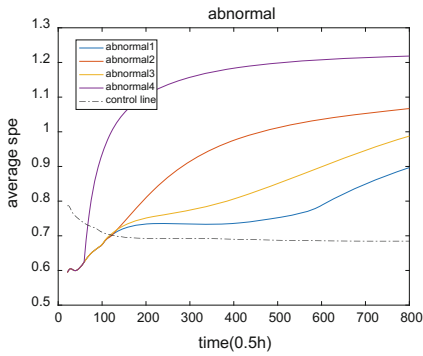
From Table 1 we can see that  $\overline{SPE}_k$  together with KPLS or AKPLS can discriminate normal batches from abnormal batches. Results in Table 2 show that AKPLS with Gaussian kernel is better than other methods because it alarms earlier than other methods in most cases after faults occur. The control charts of three methods are presented in Fig. 2, where time units are 30 minutes.

**Table 1** On-line monitoring results in penicillin dataset ( $\overline{SPE}_k$ )

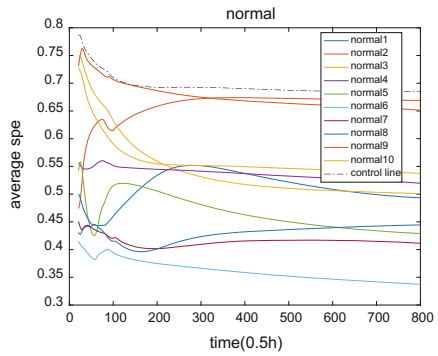
Algorithm	Number in 10 normal batches the method deems normal	Number in 4 abnormal batches the method deems abnormal
PLS	9	3
Gaussian KPLS	10	4
Gaussian AKPLS	10	4

**Table 2** Alarm time (0.5 h) of 4 abnormal batches ( $\overline{SPE}_k$ )

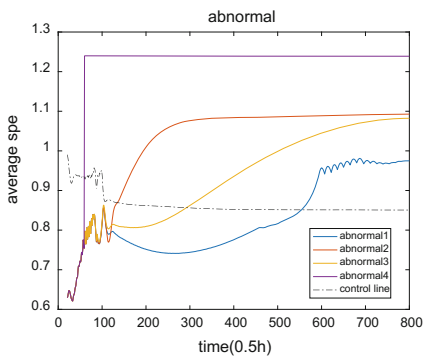
Algorithm	Fault 1	Fault 2	Fault 3	Fault 4
PLS	544	Not detected	366	62
Gaussian KPLS	554	135	293	60
Gaussian AKPLS	122	123	119	68
True start time	120	110	80	60



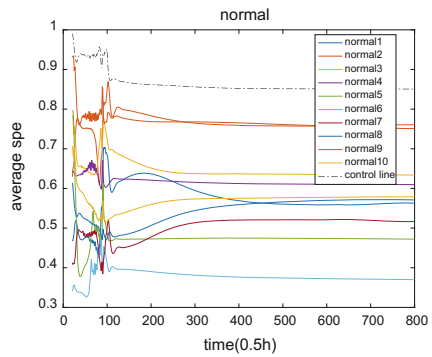
(a) abnormal batches (AKPLS)



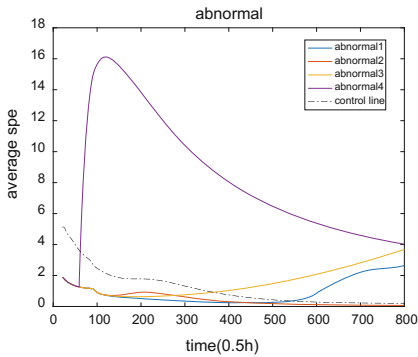
(b) normal batches (AKPLS)



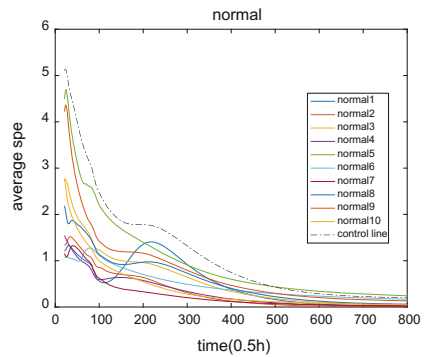
(c) abnormal batches (KPLS)



(d) normal batches (KPLS)



(e) abnormal batches (PLS)



(f) normal batches (PLS)

Fig. 2 On-line monitoring results with  $\overline{SPE}_k$



## 5 Conclusion

In this paper, we propose a special case of kernel partial least square called additive kernel partial least square (AKPLS) and show its application in batch process on-line monitoring. AKPLS uses a special kind of kernel called additive kernel where variables in different time points are mapped independently into feature space. Using ASPLS, we can estimate statistics without filling missing data after current time point in on-line monitoring. Experiments on penicillin dataset show effectiveness of AKPLS.

## References

1. Birol G, Ündey C, Cinar A (2002) A modular simulation package for fed-batch fermentation: penicillin production[J]. *Comput Chem Eng* 26(11):1553–1565
2. Ge Z, Gao F, Song Z (2011) Batch process monitoring based on support vector data description method. *J Process Control* 21(6):949–959
3. Ge Z, Song Z, Gao F (2013) Review of recent research on data-based process monitoring. *Ind Eng Chem Res* 52(10):3543–3562
4. Maji S, Berg AC, Malik J (2013) Efficient classification for additive kernel SVMs. *IEEE Trans Pattern Anal Mach Intell* 35(1):66–77
5. Nelson PRC, Taylor PA, MacGregor JF (1996) Missing data methods in PCA and PLS: score calculations with incomplete observations. *Chemometr Intell Lab Syst* 35(1):45–65
6. Nomikos P, MacGregor JF (1994) Monitoring batch processes using multiway principal component analysis. *AIChE J* 40(8):1361–1375
7. Nomikos P, MacGregor JF (1995) Multi-way partial least squares in monitoring batch processes. *Chemometr Intell Lab Syst* 30(1):97–108
8. Nomikos P, MacGregor JF (1995) Multivariate SPC charts for monitoring batch processes. *Technometrics* 37(1):41–59
9. Rosipal R, Trejo LJ (2002) Kernel partial least squares regression in reproducing kernel hilbert space. *J Mach Learn Res* 2:97–123
10. Yao M, Wang H (2015) On-line monitoring of batch processes using generalized additive kernel principal component analysis. *J Process Control* 28:56–72