PROCESS MONITORING OF AN ELECTRO-PNEUMATIC VALVE ACTUATOR USING KERNEL PRINCIPAL COMPONENT ANALYSIS

Sang-Oak Song, Gibaek Lee, En Sup Yoon

School of Chemical Engineering, Seoul National University

Abstract:. In this paper, an approach for process monitoring using a multivariate statistical technique, namely kernel principal component analysis is studied. Kernel principal analysis has recently been proposed as a new method for performing a nonlinear form of principal component analysis (PCA). The basic idea of kernel PCA is to first map the input space into a feature space via a nonlinear map and then compute the principal components in that feature space. For the process monitoring application, reconstructed input patterns can be obtained by approximating the pre-image of scores in feature space. An application study of an electro-pneumatic valve actuator in a sugar factory is described. The results show that the kernel PCA approach can detect several actuator faults earlier than linear PCA This study indicates the great potential of Kernel PCA for process monitoring.

Keywords: Kernel PCA, fault detection, actuators, control valves, process monitoring

1. INTRODUCTION

In recent process industry, on-line monitoring of process performance is extremely important for plant safety, production efficiency and product quality. As industrial systems becoming more heavily instrumented, resulting in larger quantities of data available for use in process monitoring, and modern computers are becoming more powerful, empirical modelling approaches that are basically data-driven multivariate statistical methods have attracted much interest by process engineers. These approaches are based on the theory of statistical process control (SPC), under which the behaviour of a process is modelled using data obtained when the process is operating well and in a state of control. Future unusual events are detected by referencing the measured process behaviour against this model.

Principal component analysis (PCA) is the most widely used data-driven technique for process monitoring which has been heavily studied and applied to industrial systems over the past decade. PCA is an optimal dimensionality reduction technique in terms of capturing the variance of the data, and it accounts for correlations among variables. The lower-dimensional representations of the data produced by PCA can improve the proficiency of detecting and diagnosing faults using multivariate statistics. The principal components span a low dimensional subspace used for analysis. The details of linear PCA can be found elsewhere (Jolliffe, 1986).

However, PCA is a linear technique, which ignores the nonlinearities in the process data. Industrial processes are inherently nonlinear; therefore, it may be necessary to use nonlinear methods. Kramer (1991) has generalized PCA to the nonlinear case by using autoassociative neural networks. Dong and McAvoy (1996) have developed a nonlinear PCA approach based on principal curves and neural networks that produce independent principal components (Song, 2001).

Recently, the conceptual idea of generalizing an existing linear technique to a nonlinear version by applying the kernel trick has become an area of active research. One important result in this direction is the extension of linear PCA to kernel PCA, as shown by Schölkopf, *et al*. (1998). In Kernel PCA they were not interested in principal components in input space, but rather in principal components of variables, or features, which are nonlinearly related to the input variables. Among these are for instance variables obtained by taking higher-order correlations between input variables. To this end, the method of expressing dot products in feature space in terms of kernel functions in input space is used. Given any algorithm which can be expressed solely in terms of dot products, i.e. without explicit usage of the variables themselves, this kernel method enables to construct different nonlinear versions of it (Vapnik, 1995).

The present work studies a nonlinear version of PCA using kernel technique and an application for process monitoring of an electro-pneumatic valve actuator. We first introduce the concept of Kernel PCA and reconstruction. And then Kernel PCA based process monitoring has been illustrated on the electro-pneumatic valve actuator benchmark system and its simulation results are discussed.

2. KERNEL PRINCIPAL COMPONENT ANALYSIS

1.1 Principal Component Analysis in Feature Spaces

Given a set of N centered observations $\mathbf{x}_k$, $k = 1,\cdots, M$, $\mathbf{x}_k \in \mathbf{R}^N, \sum_{k=1}^{M}\mathbf{x}_k = 0$, PCA diagonalizes the covariance matrix

$$C = \frac{1}{M}\sum_{j=1}^{M}\mathbf{x}_j\mathbf{x}_j^T \tag{1}$$

To do this, one has to solve the Eigenvalue equation

$$\lambda\mathbf{v} = C\mathbf{v} \tag{2}$$

for Eigenvalues $\lambda \geq 0$ and $\mathbf{v} \in \mathbf{R}^N \setminus \{0\}$. As $C\mathbf{v} = \frac{1}{M}\sum_{J=1}^{M}(\mathbf{x}_j \cdot \mathbf{v})\mathbf{x}_j$, all solutions $\mathbf{v}$ must lie in the span of $\mathbf{x}_1 \cdots \mathbf{x}_N$, hence (2) is equivalent to

$$\lambda(\mathbf{x}_k \cdot \mathbf{v}) = (\mathbf{x}_k \cdot C\mathbf{v}) \text{ for all } k = 1,\cdots,M . \tag{3}$$

Next, let us consider this computation in another feature space $\mathbf{F}$, which is related to the input space by a possibly nonlinear map

$$\begin{aligned}\Phi : \mathbf{R}^N &\rightarrow \mathbf{F} \\ \mathbf{x} &\rightarrow \mathbf{X}\end{aligned} \tag{4}$$

Note that $\mathbf{F}$, the feature space could have an arbitrarily large, possibly infinite, dimensionality. Here and in the following upper case characters are used for elements of $\mathbf{F}$, while lower case characters denote elements of $\mathbf{R}^N$. It is assumed that we are dealing with centered data, i.e. $\sum_{k=1}^{M}\Phi(\mathbf{x}_k) = 0$. To perform PCA in feature space, we need to find Eigenvalue $\lambda \geq 0$ and Eigenvectors $\mathbf{V} \in \mathbf{F} \setminus \{0\}$ with the covariance matrix in F,

$$\bar{C} = \frac{1}{M}\sum_{1}^{M}\Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T \tag{5}$$

Substituting $\bar{C}$ into the Eigenvector equation, we note that all solutions $\mathbf{V}$ must lie in the span of Φ-images of the training data. This implies that we can consider the equivalent system

$$\lambda(\Phi(\mathbf{x}_k)\cdot\mathbf{V}) = (\Phi(\mathbf{x}_k)\cdot\bar{C}\mathbf{V}) \text{ for all } k = 1,\cdots,M \tag{6}$$

and that there exist coefficients $\alpha_i \left(i = 1,\cdots,M\right)$ such that

$$\mathbf{V} = \sum_{i=1}^{M}\alpha_i\Phi(\mathbf{x}_i) \tag{7}$$

Combining (7) and (8), we get

$$\lambda\sum_{i=1}^{M}\alpha_i(\Phi(\mathbf{x}_k)\cdot\Phi(\mathbf{x}_i)) = \frac{1}{M}\sum_{i=1}^{M}\alpha_i(\Phi(\mathbf{x}_k)\cdot\sum_{j=1}^{M}\Phi(\mathbf{x}_j)(\Phi(\mathbf{x}_j)\cdot\Phi(\mathbf{x}_i)) \tag{8}$$

for all $k = 1,\cdots,M$

Defining an $M \times M$ matrix K by

$$K_{ij} := (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)), \tag{9}$$

this leads to

$$M\lambda K\alpha = K^2\alpha \tag{10}$$

where α denotes the column vector with entries $\alpha_1,\cdots\alpha_M$. As K is symmetric, it has a set of Eigenvectors which spans the whole space, thus

$$M\lambda\alpha = K\alpha \tag{11}$$

gives us all solutions α of Eq. (10). Note that K is positive semi definite, which can be seen by noticing that it equals

$$(\Phi(\mathbf{x}_1),\cdots,\Phi(\mathbf{x}_M))^T\cdot(\Phi(\mathbf{x}_1),\cdots,\Phi(\mathbf{x}_M)) \tag{12}$$

which implies that for all $\mathbf{X} \in \mathbf{F}$,

$$(\mathbf{X}\cdot K\mathbf{X}) = \left\|(\Phi(\mathbf{x}_1),\cdots,\Phi(\mathbf{x}_M))\mathbf{X}\right\|^2 \geq 0 \tag{13}$$

Consequently, K's Eigenvalues will be nonnegative, and will exactly give the solutions $M\lambda$ of Eq. (10). We therefore only need to diagonalizes K. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_M$ denote the Eigenvalues, and $\alpha^1,\cdots,\alpha^M$ the corresponding complete set of Eigenvectors, with λ_p being the first nonzero Eigenvalue. We normalize $\alpha^p,\cdots,\alpha^N$ by requiring that the corresponding vectors in F be normalized, i.e.

$$(\mathbf{V}^k\cdot\mathbf{V}^k) = 1 \text{ for all } k = p,\cdots,M \tag{14}$$

By virtue of (7) and (11), this translates into a normalization condition for $\alpha^p,\cdots,\alpha^M$:

$$\begin{aligned}1 &= \sum_{i,j=1}^{M}\alpha_i^k\alpha_j^k(\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) \\ &= \sum_{i,j=1}^{M}\alpha_i^k\alpha_j^k K_{ij} \\ &= (\alpha^k\cdot K\alpha^k) \\ &= \lambda_k(\alpha^k\cdot\alpha^k)\end{aligned} \tag{15}$$

For the purpose of principal component extraction, we need to compute projections on the Eigenvectors $\mathbf{V}^k$ in $\mathbf{F}$ ($k = p,\cdots,M$). Let $\mathbf{x}$ be a test point with an image $\Phi(\mathbf{x})$ in $\mathbf{F}$, then

$$(\mathbf{V}^k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) \quad (16)$$

may be called its nonlinear principal components corresponding to Φ.

1.2 The Algorithm of Kernel PCA

To perform kernel PCA, the following steps have to be carried out: first, we compute the dot product matrix.

$$K_{ij} = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij} \quad (17)$$

Next, solve (11) by diagonalizing *K*, and normalize the Eigenvector expansion coefficients α^k by requiring Eq. (15),

$$1 = \lambda_k (\alpha^k \cdot \alpha^k) \quad (18)$$

To extract the principal components (corresponding to the kernel *k*) of a test point **x**, we then compute projections onto the Eigenvectors by

$$\beta_k = (\mathbf{V}^k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i^k k(\mathbf{x}_i, \mathbf{x}) \quad (19)$$

3. RECONSTRUCTION ORIGINAL PATTERNS BY APPROXIMATE PREIMAGE

When Kernel PCA can be considered as a natural generalization of linear PCA, this can be used for data compression, reconstruction, and de-nosing applications common in linear PCA. However this is a nontrivial task, as the results provided by kernel PCA live in some high dimensional feature space and need not have pre-images in input space. Schölkopf, *et al.* (1999) presented some ideas for finding approximate pre-images.

Being just a basis transformation, standard PCA allows the reconstruction of the original patterns from a complete set of extracted principal components by expansion in the Eigenvector basis. In Kernel PCA, this is no longer possible, the reason being that it may happen that a vector **V** in **F** does not have a pre-image in $\mathbf{R}^N$. We can, however, find a vector **z** in $\mathbf{R}^N$ which maps to a vector that optimally approximates **V**.

To reconstruct the Φ-image of a vector **x** from its projections onto the first *n* principal components in **F** (assuming that the Eigenvectors are ordered by decreasing Eigenvalues size), we define a projection operator P_n by

$$P_n \Phi(\mathbf{x}) = \sum_{k=1}^{n} \beta_k \mathbf{V}^k \quad (20)$$

If *n* is large enough to take into account all directions belongs to Eigenvectors with non-zero Eigenvalue, we have $P_n \Phi(\mathbf{x}_i) = \Phi(\mathbf{x}_i)$. Otherwise Kernel PCA still satisfies that the overall squared reconstruction error $\sum_i \|P_n \Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_i)\|^2$ is minimal and the retained variance is maximal among all projections onto orthogonal directions in **F**. In common applications, however, we are interested in a reconstruction in input space rather than in **F**. To achieve this we compute a vector **z** by minimizing

$$\rho(\mathbf{z}) = \|\Phi(\mathbf{z}) - P_n \Phi(\mathbf{x})\|^2 \quad (21)$$

The hope is that for the kernel used, such a **z** will be a good approximation of **x** in input space.
In (21), replacing terms independent of **z** by Ω, we obtain

$$\rho(\mathbf{z}) = \|\Phi(\mathbf{z})\|^2 - 2(\Phi(\mathbf{z}) \cdot P_n \Phi(\mathbf{x})) + \Omega \quad (21)$$

Substituting (20) and (7) into (21), we arrive at an expression which is written in terms of dot products. Consequently, we can introduce a kernel to obtain a formula for ρ which does not rely on carrying out Φ explicitly.

$$\rho(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - 2\sum_{k=1}^{n} \beta_k \sum_{i=1}^{M} \alpha_i^k k(\mathbf{z}, \mathbf{x}_i) + \Omega \quad (22)$$

4. CASE STUDY: ELECTRO-PNEUMATIC VALVE ACTUATOR BENCHMARK PROBLEM

To verify and illustrate the usefulness of Kernel PCA for process monitoring, data generated from the control valve actuator benchmark system were used.

The actuator benchmark problem was built by Development and Application of Methods for Actuator Diagnosis in industrial Control Systems (DAMADICS) research training network for comparing the properties of fault detection and isolation methods based on the real sugar factory (DAMADICS RNT Information Website). The benchmark actuator selected is a final control element or simply named actuator, which interacts with the controlled process. The input of actuator is the output of the process controller (flow or level controller) and the actuator modifies the position of the valve allowing a direct effect on the primary variable in order to follow the flow or level set point.

Figure I shows the actuator scheme. The actuator consists in three main components: control valve, spring-and-diaphragm pneumatic servo-motor and positioner. Control valve is the mean used to prevent and/or limit the flow of fluids. Changing the state of the control valve is accomplished by a servomotor. A spring-and diaphragm pneumatic servomotor can be defined as a compressible (air) fluid powered device

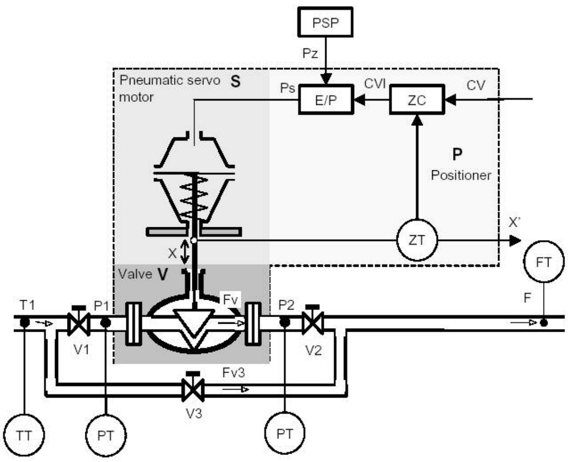

Figure. I. The actuator scheme

in which the fluid acts upon the flexible diaphragm, to provide linear motion of the servomotor system. Positioner is a device applied to eliminate the control-valve-stem miss-positions produced by the external or internal sources such as friction, pressure unbalance, hydrodynamic forces etc. It consists in a inner loop with a P controller of a cascade control structure, including the output signal of the outer loop of the flow or level controller and the inner loop of the position controller. More details are in DAMADICS RNT Information Website.

The basic measured physical values are composed of six variables: external controller output (CV), flow sensor measurement (F), valve input pressure (P1), valve output pressure (P2), liquid temperature (T1) and rod displacement (X). The Simulink library constructed by a non-linear mathematical model of the valve was used to generate faulty or fault-free data to evaluate Kernel PCA based process monitoring. All the measurement signals are normalized in the range of <0, 1> referring to the real measurement spans.

The training data for Kernel PCA model of the valve actuator system are generated without any fault for 2400 seconds. Total 2100 data except set-up zone data for initial 300s are used for building a process monitoring model.

Four kind of fault scenarios are considered for actuator monitoring in this study.

- Scenario I: Control valve faults (Valve clogging/ small abrupt fault)
- Scenario II: Control valve faults (Increased of valve or bushing friction/ incipient fault)
- Scenario III: Pneumatic servo-motor faults (Servo-motor's spring fault/ big abrupt fault)
- Scenario IV: Positioner fault (Rod displacement sensor fault/ incipient fault)

All faults are introduced at 900s of simulation time. The initial set-up zone (300s) is not also considered to avoid taking into account false detections which can occur at the beginning. Therefore, the fault situations are introduced at 600s in effect.

5. RESULTS

Two models of linear PCA and Kernel PCA are compared for verifying the potential of Kernel PCA technique. For this comparison, we define some performance indexes.

- Detection time (T_{dt}): time of detecting fault in three successions.

-True detection rate (R_{td}):

$$R_d = \frac{\textit{the number of fault detection}}{\textit{faulty situation period}} \times 100$$

We adopted the detection time for three successive detections in order to avoid taking into account false detection moments. One can consider false detection rate as one of performance indexes. In this study, we use 99% control limits and then false detection rate is

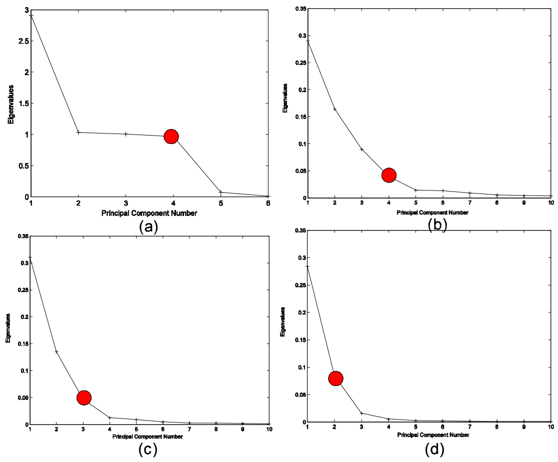

Figure. II. The eigenvalues plot

(a) Linear PCA
(b) Kernel PCA (σ^2=0.1)
(c) Kernel PCA (σ^2=0.2)
(d) Kernel PCA (σ^2=0.4)

very small (almost zero). Thus, this index is excluded for the comparison.

The number of principal components (PCs) to retain in the model should be determined for both of PCA and Kernel PCA before training. In the case of Kernel PCA, we should determine the Kernel type and corresponding parameters (e.g. bandwidthin the case of RBF Kernels). We used RBF Kernels in this work. Some research shows that RBF Kernels consistently yield good performance through an empirical assessment of Kernel type performance (Baesens, *et. al*., 2000).

In general, the choice of the number of PCs in standard PCA is made by cross validation, a few rules of thumb and the user's knowledge of the data. 4 PCs (98.59 % variance captured) are selected from Figure. II. (a) in this work. It is generally useful to plot the eigenvalues. When looking at the plot of eigenvalues, one looks for a sudden jump in the values from the small ones. In the Kernel PCA, the problem how many principal components are used depends on the Kernel parameters determined (σ^2). Figure. II. (b)-(d) shows that the lager parameter one use, the smaller PCs one should choose. We can understand this relation intuitively from the fact that RBF Kernels with larger bandwidth can capture more complex features. By cross validation, we determined the RBF Kernels with σ^2=0.1 and 4 PCs (about 92% variance captured) to retain in the Kernel PCA model.

When using PCA, one uses primarily Q and T^2 for detecting system faults. Q statistic is a measure of the variation in the data outside the PCA model. T^2 statistic, on the other hand, is a measure of the distance from the multivariate mean to the projection of the operating point onto the hyper plane defined by the PCs, that is, a measure of the variation within the PCA model. In practice, violations of the Q and T^2 limits generally occur for different reasons. Assuming a normal value of Q, a T^2 fault indicates that the process has gone outside the usual range of operation but in a direction of variation common to

Table. 1 The comparison of performance indexes

Scenarios	Detection Time (s)		True Detection Rate (%)	
	Linear PCA	Kernel PCA	Linear PCA	Kernel PCA
I	617	611	62.73	51.60
II	2960	2959	19.60	13.79
III	-	-	3.40	0.13
IV	761	692	79.95	76.60

the process. A Q fault indicates that the process has gone in an entirely new direction-something entirely new has happened. Most process faults show up in Q. Very few faults are detected by T^2 alone (Wise, B.M., *et al*., 1999). In this work we use only Q statistics and 99% control limit for monitoring measure because T^2 statistics are under control limit about all fault scenarios.

Next Figures and Table 1 summarize the simulation results. They show that Kernel PCA outperforms linear PCA about all fault scenarios. In the case of scenario III, servo-motor's spring fault doesn't affect measured variables much and both of two models can not detect this fault well. We can not obtain the performance index of detection time in this fault scenario. However, the true detection rate of Kernel PCA model is much larger than one of linear PCA in scenario III.

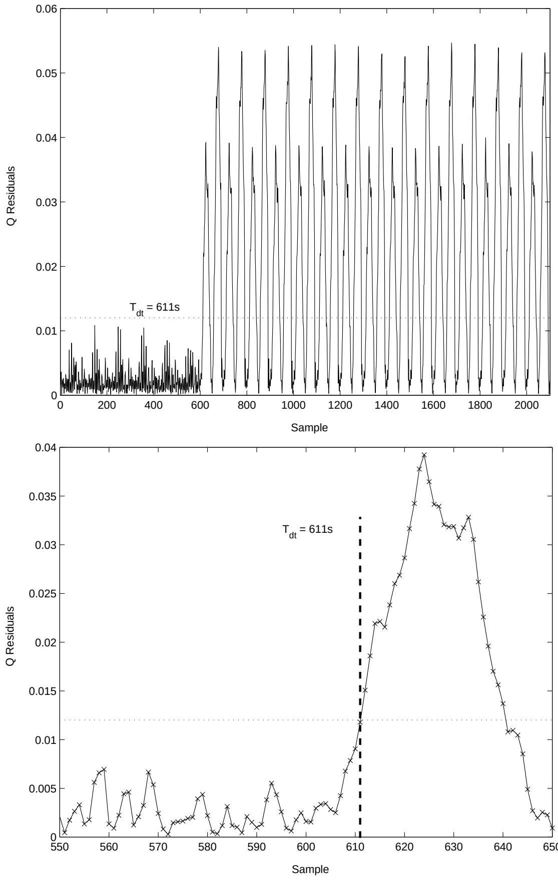

Figure. IV. The Kernel PCA monitoring result of scenario I

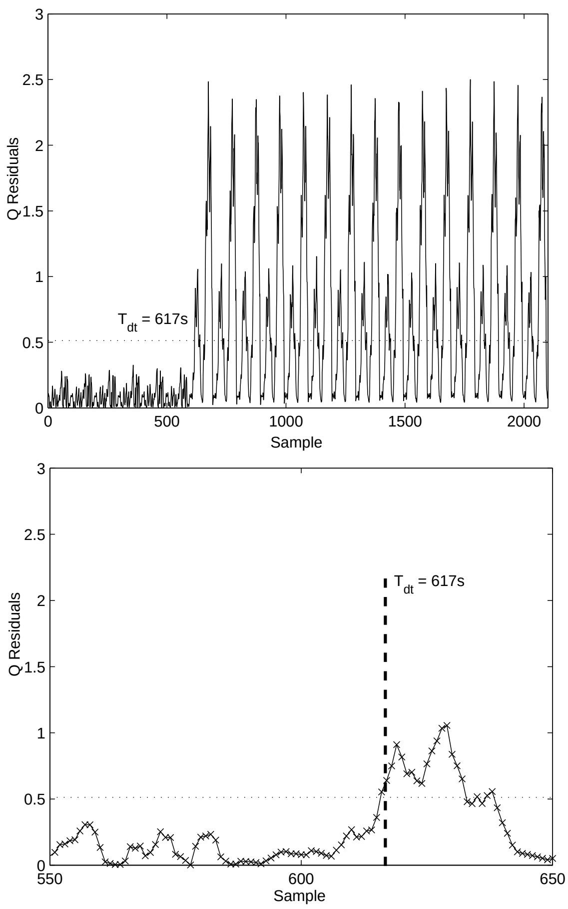

Figure. III. The PCA monitoring result of scenario I

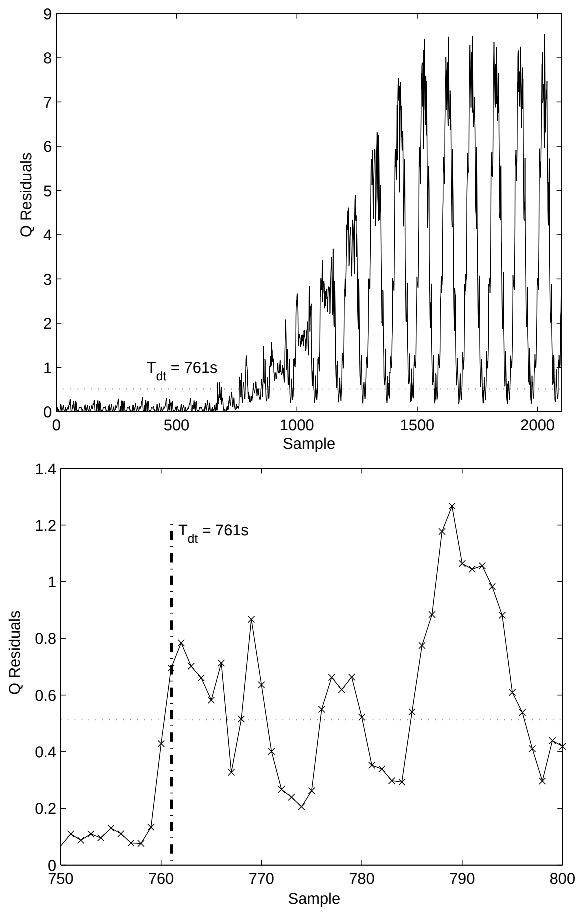

Figure. V. The PCA monitoring result of scenario IV

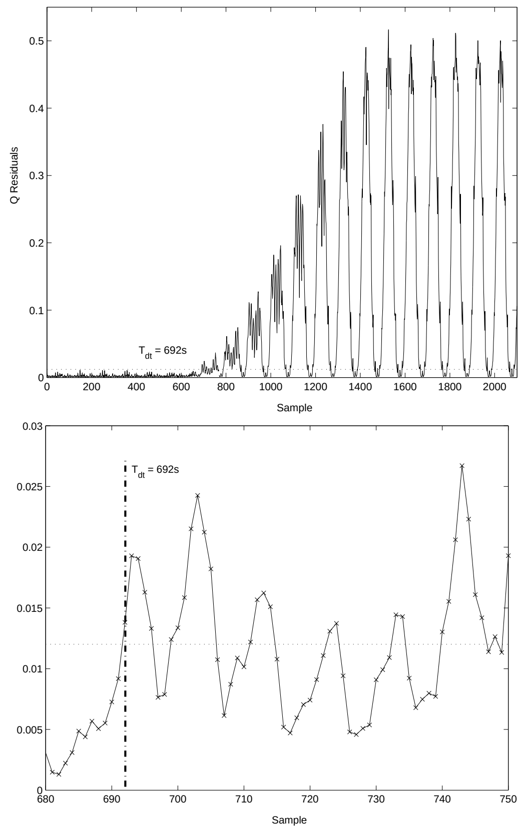

Figure. VI. The Kernel PCA monitoring result of scenario IV

4 PCs of linear PCA can captures almost 99% of variance. Thus One can assert that this actuator system can be modelled using linear PCA sufficiently and nonlinear technique such as Kernel PCA does not have great advantage against linear PCA. However Figure. VII shows the small nonlinearity in the training data. This nonlinearity makes the performance differences between Kernel PCA and linear PCA. If we apply Kernel PCA to more complex and nonlinear systems (e.g. some polymerization processes or biochemical processes), the monitoring performance will be much better.

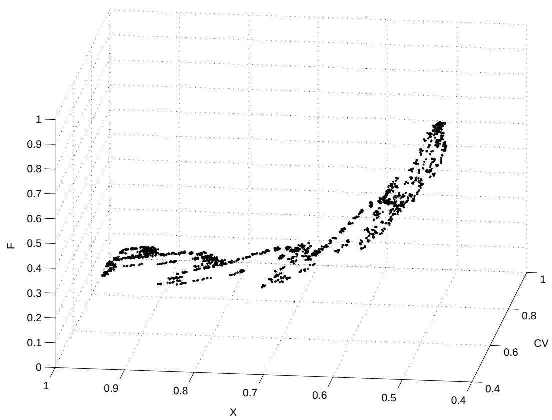

Figure. VII. The nonlinearity of normal training data

6. CONCLUSION

Kernel PCA can be considered as a nonlinear version of PCA and extract more information in nonlinear systems. However, Kernel PCA dose not provide the exact reconstructed input patterns due to implicit mapping procedure to high dimensional feature space and have some restriction on applying to process monitoring.

In this work, we reconstruct input patterns by approximating pre-images and apply to valve actuator fault monitoring. The simulation result shows that Kernel PCA based monitoring can detect several actuator faults better and earlier than conventional PCA based one. As real world industrial processes are not linear clearly, the process monitoring approach using Kernel PCA has great potential to fault diagnosis of the industrial processes.

ACKNOWLEDGEMENT

We acknowledge the financial aid for this research provided by the Brain Korea 21 Program supported by the Ministry of Education. In addition, we would like to thank the Automation & Systems Research Institute and Research Institute of Engineering Science of Seoul National University.

REFERENCES

Baesens, B., Viaene, S., Van Gestel, T., Suykens, J.A.K., Dedene, G., De Moor, B. and Vanthienen, J. (2000), An empirical assessment of Kernel type performance for least squares support vector machine classifiers, *Proceeding of 4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, 313-316.

Dong, D. and McAvoy, T.J. (1996), Nonlinear principal component analysis based on principal curves and neural networks, *Computers & Chemical Engineering*, **20**, 65-78.

Jolliffe, I.T. (1986), *Principal Component Analysis*, Springer Verlag, New York.

Kramer, M.A. (1991), Nonlinear principal analysis using autoassociative neural networks, *AIChe J.*, **37**, 233-243.

Schölkopf, B., Smola, A. and Müller, K. (1998), Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, **10** (5), 1299-1399.

Schölkopf, B., Mica, S, Burges, C.J.C., Knirsch, P., Müller, K, Rätsch, G. and Smola, A.J. (1999), Input Space Versus Feature Space in Kernel-Basd Methods, IEEE transactions on neural networks, **10**(5), 1000-1017.

Song, S.O., Yoon, E.S. and Chang, K.S. (2001), Analysis of novelty detection properties of autoassociators, *Proceeding of '2001 China-Korea Joint Workshop on PSE*, 1-5.

Wise, B.M., Gallagher, N.B., Butler, S.W., White JR, D.D. and Barna, G.G. (1999), A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process*, J. Chemometrics*, 13, 379-396.

Vapnik, V.(1995), *The Nature of Statistical Learning Theory*, Springer Verlag, New York.