N° Ordre………./IGEE/UMBB/2021

PEOPLE’E DEMOCRATIC REPUBLIC OF ALGRIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

M’HAMED BOUGARA UNIVERSITY OF BOUMERDES

Institute of Electrical and Electronic Engineering

Ph. D. Dissertation

Presented by:

BENCHEIKH Fares

In Partial Fulfillment of the Requirement for the Degree of

DOCTORATE

Field: Automatique

Option : Automatique

Title:
Multivariate Statistical Process Monitoring Using Kernel Principal Component Analysis

Jury Members:

•	Mr	Abdelhakim	KHOUAS	Prof	University M’hamed BOUGARA Boumerdes, Algeria	Jury President
•	Mr	Mohammed Faouzi	HARKAT	Prof	University BADJI Mokhtar Annaba, Algeria	Thesis Director
•	Mr	Abdelmalek	KOUADRI	Prof	University M’hamed BOUGARA Boumerdes, Algeria	Thesis Co-director
•	Mr	Chemseddine	RAHMOUNE	MCA	University M’hamed BOUGARA Boumerdes, Algeria	Examiner
•	Mr	Abderrazak	LACHOURI	Prof	University 20 Août 1955 Skikda, Algeria	Examiner

Academic year: 2020/2021

ABSTRACT

Fault detection and diagnosis field (FDD) plays an important role in industrial processes. It assures the safe operation of the process and reduces its maintenance costs. The implementation of mechanisms for early detection and diagnosis of faults is called process monitoring. Due to the size and complexity of industrial processes, multivariate statistical methods are finding wide application in process monitoring. Some popular methods are principal component analysis (PCA) for linear processes, and kernel principal component analysis (KPCA) for nonlinear processes.

The main challenge in the KPCA based fault detection and diagnosis method is **the high computation time and memory storage space** whenever the size of the training data increases. The developed kernel matrix size depends on the number of training observations. So, it requires $O(n^2)$ storage space for its build and for which $O(n^3)$ computation time for its eigendecomposition procedures.

In this dissertation, three new methods have been proposed to address the computation drawbacks of KPCA. The first method aims to eliminate the redundant observations among the training dataset based on the Euclidean distances between observations such that any two observations with zero Euclidean distance are considered similar and one of them can be removed. The second method removes the correlated observations and keeps only the representative non-correlated observations to build a reduced training dataset. The third method reduces the training dataset by eliminating the dependent observation guarding only the independent observations. The reduced training datasets are used to build KPCA algorithm to compute the fault indices thresholds in order to fire the alarms when the index violated its threshold. The proposed methods are applied to two case study industrial processes: Ain El Kebira rotary kiln process and Tennessee Eastman process. The obtained results are compared to the ordinary KPCA and different Reduced KPCA (RKPCA) methods; in terms of false alarm rate (FAR), missed detection rate (MDR), and detection time delay (DTD); to evaluate the efficiency of these proposed methods. The proposed RKPCA techniques are able to enhance the time and space computation of KPCA and contribute better monitoring performance.

Keywords: Fault detection, Principal component analysis, Kernel PCA, Reduced KPCA, Redundancy, Euclidean distance, Correlation, Independence, Cement rotary kiln, Tennessee Eastman process, Computation time, Computation space

RESUME

Les méthodes de détection et de diagnostic de défauts sont importants dans les processus industriels. Elles assurent le fonctionnement du processus et réduisent ses coûts de maintenance. La mise en œuvre de mécanismes de détection et de diagnostic de défauts est appelée surveillance de processus. En raison de la taille et de la complexité des processus industriels, les méthodes statistiques multi-variées trouvent une large application dans la surveillance de processus. Les méthodes les plus populaires sont l'analyse en composantes principales (ACP ou PCA en anglais pour principal component analysis), pour les processus linéaires et l'analyse des composants principaux du noyau (KPCA) pour ceux non linéaires. Les défauts sont détectés avec des indices de détection de défaut qui déclenchent des alarmes lorsqu'un index a dépassé sa limite de contrôle. Le principal défi de la méthode de détection et de diagnostic de défauts basée sur KPCA est le temps de calcul et l'espace de stockage élevés lorsque la taille des données d'apprentissage augmente. La taille de la matrice du noyau développée dépend du nombre d'observations d'apprentissage. Ainsi, elle nécessite $O(n^2)$ d'espace de stockage pour sa construction et pour laquelle $O(n^3)$ de temps de calcul pour ses procédures d'eigen-décomposition.

Dans cette thèse, trois nouvelles méthodes ont été proposées pour traiter ces problemes. La première méthode vise à éliminer les observations redondantes dans l'ensemble des données d'apprentissage en se basant sur les distances euclidiennes entre les observations, de sorte que deux observations ayant une distance euclidienne nulle sont considérées comme similaires et l'une d'entre elles peut être éliminée. La deuxième méthode supprime les observations corrélées et ne conserve que les observations représentatives non corrélées pour construire un ensemble de données d'apprentissage réduit. La troisième méthode réduit l'ensemble de données d'apprenti- ssage en éliminant les observations dépendantes et en ne conservant que les observations indépendantes. Les ensembles de données d'apprentissage réduits sont utilisés pour construire l'algorithme KPCA afin de calculer les seuils des indices de défaut et de déclencher les alarmes lorsque l'indice viole son seuil. Les méthodes proposées sont appliquées à deux cas d'étude de processus industriels : Le processus de four rotatif d'Ain El Kebira et le processus de Tennessee Eastman. Les résultats obtenus sont comparés à ceux de la méthode KPCA ordinaire et de différentes méthodes RKPCA, en termes de taux de fausses alarmes (FAR), de taux de détection manquée (MDR) et de délai de détection (DTD), afin d'évaluer l'efficacité des méthodes proposées. Les techniques RKPCA proposées sont capables d'améliorer le calcul en temps et en espace de la KPCA et contribuent à une meilleure performance de surveillance.

Mots clés: Détection et de Diagnostic de Défauts, analyse en composantes principales, Analyse en composantes principales du noyau, distances euclidiennes, Corrélation, Indépendance, four rotatif à ciment, Tennessee Eastman, Temps de calcul, Espace de calcul.

ملخص

يلعب ميدان اسكتشاف الأخطاء و تشخيصها دورا هاما في المجالي الصناعي, فهو المسوؤل عن التشغيل الآمن للعمليات الصناعية و التقليل من تكاليف التشغيل و الصيانة. يطلق على بناء آليات استكشاف الاخطاء المبكرة و تشخصيها "مراقبة العمليات الصناعية" حيث تعتمد أساسا على الأساليب الإحصائية المتعددة, من أبرز الأساليب الاحصائية الشائعة هي تحليل المركبات الرئيسية (PCA) للعمليات الخطية ، و تحليل االمركبات الرئيسية للنواة (KPCA) للعمليات غير الخطية. في مراقبة العملية الإحصائية. يتمثل العائق الرئيسي في طريقة اكتشاف الأخطاء وتشخيصها المستندة إلى KPCA في وقت الحساب ومساحة تخزين الكبيران كلما زاد حجم بيانات التدريب. حيث يعتمد حجم مصفوفة النواة المطورة على عدد عينات التدريب. لذلك يتطلب الأمر خوارزمية بمساحة تخزين$O(n^2)$ و وقت حساب $O(n^3)$ لإجراءات التحليل الذاتي.

في هذه الأطروحة تم اقتراح ثلاث طرق جديدة لمعالجة المشكل المطروح. تهدف الطريقة الأولى إلى حذف العينات المتشابهة بين مجموعة بيانات التدريب بناًء على المسافات الإقليدية, حيث تعتبر أي عينتين بمسافة إقليدية صفرية متشابهة ويمكن إزالة واحدة منهما. أما الطريقة الثانية فتزيل العيننات المرتبطة وتحتفظ فقط بالعيينات التمثيلية الغير مرتبطة لإنشاء مجموعة بيانات تدريب مصغرة. في الطريقة الثالثة يتم حذف العيينات الغير مستقلة احصائيا من مجموعة بيانات التدريب و الحفاظ على المستقلة فقط. تُستخدم مجموعات بيانات التدريب المصغرة لبناء خوارزمية KPCA لحساب عتبات مؤشرات الكشف عن الأخطاء من أجل إطلاق الإنذارات عندما يتجاوز أحد الموشرات العتبة الخاص به. لتقييم كفاءة هاته الطرق المقترحة تم تطبيقها على عمليتيبن دراسيين صناعيتين: عملية فرن انتاج الاسمنت في عين الكبيرة وعملية Tennessee Eastman ومقارنتها بطرق RKPCA المختلفة.

الكلمات المفتاحية: اسكتشاف الأخطاء و تشخيصها, تحليل المركبات الرئيسية, و تحليل االمركبات الرئيسية للنواة, وقت الحساب, ومساحة تخزين, العينات المتشابهة, المسافات الإقليدية, العيننات المرتبطة, العيينات المستقلة, عملية فرن انتاج الاسمنت, عين الكبيرة, عمليةTennessee Eastman.

ACKNOWLEDGEMENT

Praise and glorification be only to Allah, the Almighty, the most beneficent and the most merciful, whose blessing and guidance have helped me finish my PhD thesis.

With immense pleasure and deep sense of gratitude, I would like to express my sincere gratitude to my supervisor **Pr. HARKAT Mohammed Faouzi** *without his motivation and continuous encouragement, inspiration, and guidance this research would not have been successfully completed.*

My profound sense of gratitude to my co-supervisor **Pr. KOUADRI Abdelmalek** *for his continuous support and advice during my research. His guidance was of a great help in all the time of research and writing of this thesis.*

I would also like to thank ***my Ph.D mates: ROUANI Lahcene, AYACHI AMOR Yacine, RABIAI Zakaria, GRAINAT Youcef*** *for their help and encouragement.*

I would also like to acknowledge the help and support of ***my beloved family*** *members and* ***my friends*** *for their constant encouragement and moral support with patience and understanding.*

BENCHEIKH Fares

DEDICATION

To my beloved parents,
To my brothers “Omar, Khalil, Taissir, and Wassim ”,
To my two sisters “Lina and Tasnim”,
To extended family,
To my best friends,
To those who will be happy for me.

Table of Contents

TABLE OF CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

List of Acronyms and Abbreviations

CPV	Cumulative Percent of Variance
CVA	Canonical Variate Analysis
DTD	Detection Time Delay
ED	Euclidean Distance
EM	Expectation Maximization
FAR	False Alarm Rate
FDA	Fisher Discriminant Analysis
FDD	Fault Detection and Diagnosis
ICA	Independent Component Analysis
KCCA	Kernel Canonical Correlation Analysis
KFD	kernel Fisher discriminant analysis
KLD	Kullback-Liebler Divergence
KPCA	Kernel Principal Component Analysis
PM	Process Monitoring
PCA	Principal Component Analysis
MDR	Missed Detection Rate

PLS	Partial Least square
PC	Principal Component
RDKPCA	Reduced Dynamic KPCA
RIDKPCA	Reduced Interval-valued Dynamic KPCA
RIKPCA	Reduced Interval-valued KPCA
RKPCA	Reduced Kernel Principal Component Analysis
SFA	Slow Feature Analysis
SPE	Squared Prediction Error
SVD	Singular Value Decomposition
TEP	Tennessee Eastman Process

List of Symbols

X	Training data matrix
x	Measurement vector
m	Number of variables
n	Number of samples
r	Number of samples of reduced dataset
C	Covariance matrix
Λ	Diagonal eigenvalues matrix
P	Eigenvectors matrix
l	Number of retained principle components
ϕ	Feature space mapping function
K	Kernel matrix
σ	Kernel parameter
α	Kernel matrix eigenvectors
V	Feature space eigenvectors
K_r	Reduced Kernel matrix
T^2	Hotelling's index
Q	Squared prediction error
φ	Combined index
T	Score matrix
T^2_α	Hotelling's index threshold
Q_α	Squared prediction error threshold
φ_α	Combined index threshold

β	Euclidean distance redundancy threshold
γ	Correlation threshold
$J(.,.,.)$	Loss function
$O(.)$	Big O (Time/space complexity)

1 General Introduction

1.1 Background

In the last century, The field of fault detection and diagnosis (FDD) has seen an increasing emergence due to the high development of modern industrial processes, demands for product quality, and operation safety. So, it is required to build sophisticated and advanced techniques that can quickly and correctly detect abnormalities and determine whether the process is under healthy or faulty operation mode. The instantaneous detection of faults ensures enough warning time for the fault diagnosis scheme to identify the source location of the detected faults that occurred in the process. The quick detection and diagnosis of the abnormalities prevent deterioration of the process behavior and avoid any long period reparations. These techniques are classified into model-based and history-based fault detection methods [1–3].

Model-based methods are heavily dependent on an explicit mathematical model of the process and the analytic relations between the inputs and outputs to extract information about the causes of faults [4–6]. Generally, these techniques are limited to small processes, which their accurate mathematical model can be easily determined. In large-scale and complex industrial processes, it is a challenging issue to acquire an accurate mathematical process model [7]. Model-based methods mainly use Kalman filter [8], and state estimation [9] to detect the abnormalities through generated residuals.

In the model-based techniques the high interactions between the variables, among time-varying parameters and nonlinearities in large-scale industrial processes prevent these from obtaining good monitoring performance [10]. On the other hand, the huge advancement in measurement equipment and data storage materials gives data-driven methods the advantage to monitor modern processes. They make use of process historian data to construct an explicit model of the normal operation mode and detect by divergence the abnormalities and their root [3]. These techniques reduce the dimensionality of the process data using machine learning techniques along with multivariate statistical methods to construct a comprehensive model for the process.

Some well-known statistical techniques are Principal Component Analysis (PCA), Partial Least Squares (PLS), Fisher Discriminant Analysis (FDA), and Independent Component Analysis (ICA) [2, 11, 12]. Principal component analysis (PCA) was proposed in 1901 by Pearson [13] and developed by Hotelling [14]. It is a linear transformation that projects the input data onto a new lower dimension space that captures the most effective variations in the data.Since PCA is a linear technique, therefore, it does not take into consideration the nonlinear relations revealed in industrial processes [15]. Thus, the monitoring performance of industrial processes with high nonlinear correlations between variables degrades when PCA technique is applied. Many techniques have been proposed to solve that issue. Kernel PCA (KPCA) as a nonlinear generalization of the PCA is proposed by Scholkopf et al. in [16]. KPCA maps the original dataset into a higher dimension (or infinite) feature space, afterwards, standard PCA is performed in that feature space. Unlike Neural network based techniques, KPCA solves an eigenvalue problem instead of solving a nonlinear optimization problem [17, 18]. Process monitoring based on KPCA technique uses the same PCA fault detection indices such as the Hotelling's T^2 and Q statistics. The main challenge in the KPCA based fault detection and diagnosis method is the high computation time and memory storage space whenever the size of the training data increases. The developed kernel matrix size depends on the number of training observations. So, it requires $O(n^2)$ storage space for its build and for which $O(n^3)$ computation time for its eigen-decomposition procedures [19, 20]. To address these evidently related issues, many techniques have been proposed. In [20], a new Reduced KPCA (RKPCA) is proposed, this method uses singular value decomposition (SVD) to reduce KPCA. (SVD-RKPCA) method is proposed for online monitoring of nonlinear processes. The SVD-RKPCA method consist of two phases (offline and online). First It aims to find a reduced dataset by selecting the variables with the highest projection variance to build an initial KPCA model(offline phase).Then it updates the model by applying the SVD-KPCA technique online. A K-means clustering method is used to extract a reduced number of training observations [21]. Although, it is able to reduce the computation cost, however, it requires the number of clusters to be defined in advance. Nevertheless, it doesn't take into consideration the variation of parameters, so it may lead to monitoring errors. Rosipal et al. have utilized expectation maximization (EM) algorithm to solve the computational issue of KPCA although EM algorithm convergence is not guaranteed to KPCA technique [22]. A kernel Hebbian algorithm was proposed by Kim et al. which iteratively estimates the kernel principal components with only linear order memory complexity [23].

Zheng et al. have improved a version of KPCA based on the eigenvalue decomposition of a symmetric matrix. In this technique, the training dataset is divided into different subsets, where each of them is handled separately. The method improves the time cost but it consumes more memory space [24]. Zhang et al. have addressed the problem utilizing techniques from stochastic optimization to solve kernel PCA with linear space and time complexities per iteration [19].

1.2 Objectives

Since a large number of training observations lead to high time and space-consuming algorithms and prevent the quick convergence of machine learning algorithms, removing irrelevant observations could provide a computationally effective solution and may lead to an improvement in the performance of the algorithm [10, 25]. Observation reduction is a crucial step for accelerating KPCA model building without losing the process monitoring performance. After the development of the big data field and high storage computers, the number of observations collected is often massive thus dimension reduction techniques are becoming more and more imperious to build an acceptable model with high accuracy. The temptation to build a KPCA monitoring model using all collected observations is becoming too hard due to high computational complexities. It is carefully required to reduce the number of observations. The main objective of this thesis is to proposed new three algorithms to reduce the training dataset size by removing the irrelevant observations and keep the most relevant samples that can monitor the process with the same quality as the entire training dataset.

1.3 Outline

Chapter 2 gives a background on the field of fault detection and diagnosis, It provides a description of the different fault detection techniques (model-based and data-driven). A detailed explanation of data-driven is presented as well as the fault indices used to detect whether the process is under a healthy or faulty state. In Chapter 3, PCA and KPCA methods are well presented. it gives the mathematical explanation of these techniques as well as the notation used in their derivation. furthermore, it shows the limitations of PCA and KPCA. where PCA cannot handle the processes that reveal nonlinear characteristics because PCA assumes that the process variables are linear correlated. meanwhile, KPCA struggles with high time and space complexity to monitor the process with the high number of observations. Chapter 4 presents the proposed Reduced KPCA algorithms used to solve the high time and space consumption in KPCA. Three methods have been presented to reduced the training dataset size. In chapter 5, The proposed RKPCA techniques are validated and tested on two industrial processes. Ain El Kebira rotary kiln and Tennessee Eastman. The results of the proposed RKPCA are summarized in tables and compared to KPCA, and recently published RKPCA algorithms. The dissertation is closed with a general conclusion where it summarizes all the discussed techniques and the obtained results using these techniques to monitor industrial processes. In addition, the future work to develop the proposed techniques and extend RKPCA techniques to monitor more processes and enhance the performance.

2

Literature review

2.1 Introduction

The thriving field of FDD and process monitoring (PM) is crucial to guarantee the good performance and secure operation of industrial processes. During the last decade, high advances have been made both in theory and application aiming to solve the problem of process monitoring and fault detection utilizing newly developed tools involving algebraic conceptions with probabilistic and statistical techniques [26]. This chapter presents the different FDD techniques that have been developed and the FDD classes (model-based and data-driven). Data-driven is more detailed mentioning the different statistical techniques (PCA, PLS, ICA....).

2.2 Fault detection and diagnosis: State of art

PM plays an essential role in industrial processes, it assures the product quality and the proper operation and reduces the damage of the processes. Safe operation of complex industrial processes, such as oil and gas processes that demand sophisticated monitoring of process variables to improve the productivity of these processes and, more crucially, to avoid any catastrophe on any occasion of failure [27]. The world has seen a lot of catastrophic incidents that have taken place in the past few decades in various chemical and petrochemical plants. Some of these disastrous accidents are the Union Carbide accident [28, 29], the Piper Alpha accident [30, 31], and the Al-Ahmedi (Kuwait) accident [32]. In 1984, The Union Carbide accident happened in India, where a leak of toxic gas caused over 3000 deaths and 400,000 injuries [28, 29]. In 1988, 167 men were killed in Piper Alpha (an oil production plant operated by Occidental Chemical in the North Sea) accident, leaving only 61 survivors [30, 31]. Mina Al-Ahmedi accident in 2000 was caused by a failure in a condensate line in a refinery plant leading to 5 deaths and 50 injuring. The catastrophic accidents make the monitoring of the industrial process intrinsic key to avoid all these consequences and guarantee the safety of the humans and keep the profitable operation of these plants [33].

Faults can occur in open as well as closed-loop controlled systems, and they have different types, magnitudes, and behaviors over time. Additive faults [34, 35] correspond to unknown inputs acting on the process, those disturb the process outputs and are independent of the known inputs. While multiplicative faults [36–38] also known as parametric faults are variations in some process parameters which affect the process outputs depending on the magnitude of the known inputs [39]. In case of additive faults:

$$Y(t) = Y^*(t) + \Delta Y(t) = Y^*(t) + f(t) \tag{2.1}$$

and in case of multiplicative faults:

$$Y(t) = (p + \Delta p(t))U(t) = Y^*(t) + f(t)U(t) \tag{2.2}$$

Where $Y^*(t)$ is the free fault variable, $U(t)$ is the input variable, $Y(t)$ is the faulty variable, and $f(t)$ is the involuntary fault

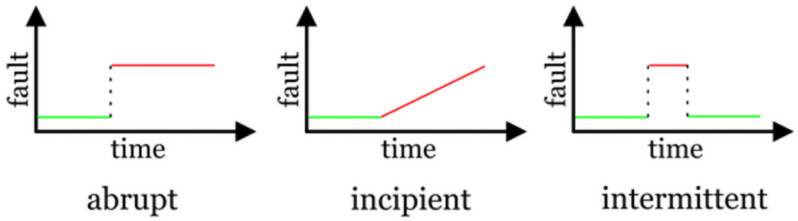


Fig. 2.1 Types of faults in terms of behavior over time

In terms of their behavior over time, faults are generally classified into three types. Abrupt faults [40–42] occur suddenly with a step-wise constant amplitude, these faults can occur as a fixed bias or a random variation and can remain permanent for the rest of the system operation or disappear after a certain transient stage. Incipient faults [43–45] on the other hand start with lower negligible amplitudes which gradually increase over time, these faults must be detected before they evolve towards their critical levels and reach other process zones. Finally, intermittent faults [46–48] pose different characteristics and challenges since they occur and vanish suddenly with different amplitudes during different time stages. Figure 2.1 displays the different fault types.

Fault detection is an important field in process monitoring. Faults in industrial processes can occur due to malfunctioning sensors or to abnormal changes in the process [3]. Sensor faults are usually caused by quick changes in a small number of process variables. on the other hand, process faults are abnormal changes due to deviations in the process itself. These faults are usually appraised by slow drifts across several variables. The monitoring techniques should be accurate and quick in detecting abnormalities. Over the last decades, many researchers have developed different FDD techniques [31, 32, 49, 50]. Generally, These fault detection techniques can be labeled into two major categories: Model-based techniques and data-driven techniques, Figure 2.2 presents the fault detection and diagnosis techniques classes.

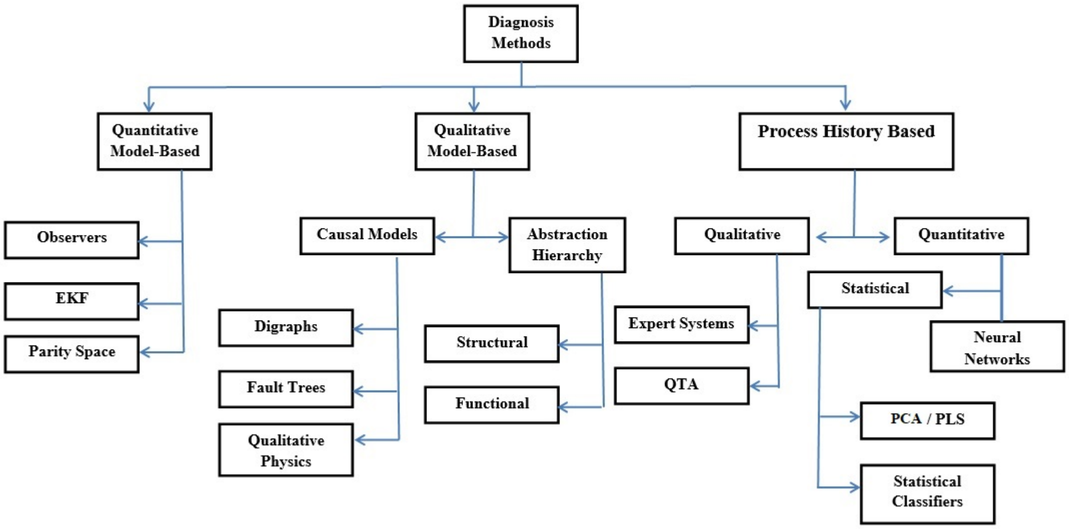


Fig. 2.2 Classification of Diagnostic Algorithms.

2.2.1 Model-based fault detection

Model-based techniques highly depend on the process model and the relationship between variables in order to identify the abnormalities. Model-based methods use analytical relations between system inputs and outputs to extract information about possible faults that may occur. Model-based uses residuals, which are the differences between the measured and the model predicted value, these residuals are used as an indicator of the existence or absence of faults [51, 52]. At normal conditions, the residuals indicate zero or close to zero in cases of uncertainties or noise. On the other hand, the significant deviation of the residuals from zero indicating the presence of a new condition that is distinguishable from the normal faultless mode [51, 52]. The main model-based monitoring approaches are the observer-based methods [53, 54], parity space approaches [55–57], and interval approaches [58]. Figure 2.3 represents a schematic diagram of model-based fault detection.

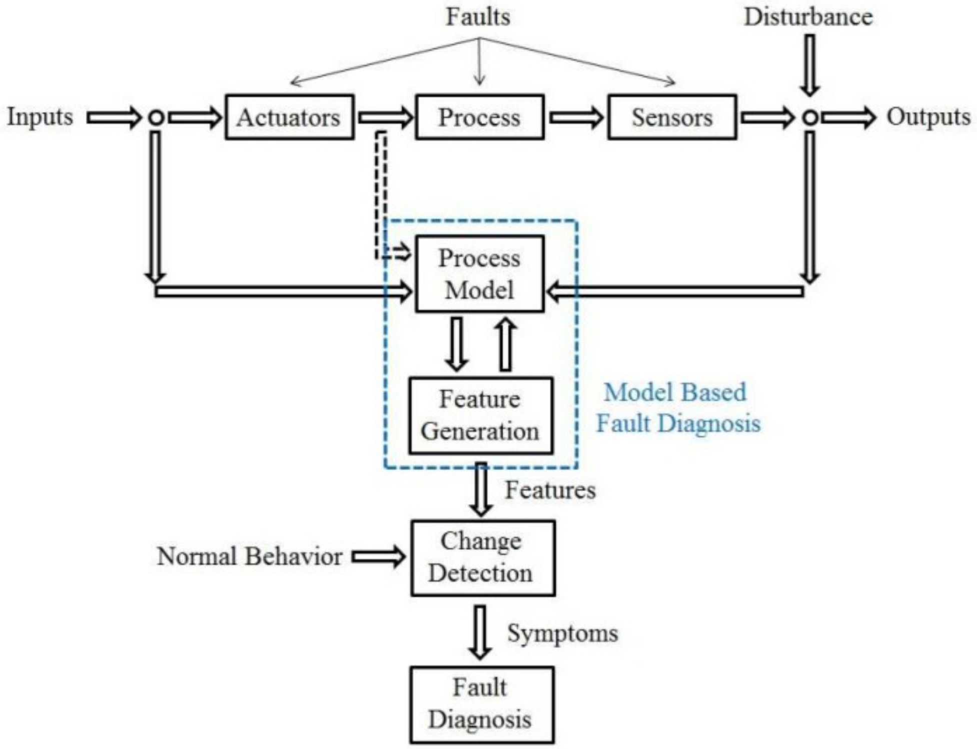


Fig. 2.3 A general schematic of model-based fault detection.

2.3 Data-driven fault detection

For the case of large-scale industrial processes, building a complete and accurate process model tends to be unfeasible which hence limits the performance of model-based FDD methods making analytical redundancy techniques applicable only for small to medium-size systems. Meanwhile, hardware redundancy methods pose many burdens in terms of occupied space, increased complexity, and financial costs; Furthermore, a common fact is that redundant components are also subject to faults and thus necessitate extra periodic maintenance efforts. Signal processing based methods as well prove to be applicable in modern systems but they are also restricted to few types of faults being known in some common systems; Major limitations of signal-based methods are due to the required dependency of the monitored index on its quality relevant variables or the studied fault on its corresponding performance indicators. Making superior alternative solutions to the previously mentioned techniques, data-driven FDD methods started to become more popular since the field of multivariate statistical process control (MSPC) was introduced in the

1990s. In this direction, several methods are proposed in the literature and proved efficient in FDD and PM in theoretical and experimental applications. Diverse methods are capable of handling multivariate big data of different types and distributions, and several extensions are further proposed.

Data-driven techniques use historical process data in order to extract the main features of the process utilizing multivariate statistical techniques. Multivariate statistical techniques are considered powerful tools that can compress data and reduce dimensionality to retain essential information that is easier to be analyzed. they are also can handle noise and correlation effectively extracting true information. PCA is based on an orthogonal decomposition of the covariance matrix of the process variables along with directions that explain the maximum variation of the data. The main purpose of using PCA is to find factors that have a much lower dimension than the original data set which can properly describe the major trends in the original data set.

PCA is a procedure used for a single data matrix, e.g., the matrix of the process variable X. Oftentimes we also have an additional group of data, e.g., product quality variables Y PLS method models the relationship between two blocks of data while compressing them simultaneously. It is used to extract latent variables that not only explain the variation in the process data X but also that variation in X is most predictive of the quality data Y. The first PLS latent variable is the linear combination of the process variables that maximizes the covariance between them and the quality variable.

FDA [59–63] method on the basis of classification and discrimination among classes; Canonical variate (Correlation) analysis (CVA) [64–66]; and slow feature analysis (SFA) [67–70] method that extracts the slow features according to their invariant levels and used to provide representations for process operation during steady and dynamic states. These methods have recently drawn increasing attention for their reliability and good performance in PM and FDD. Theoretical and experimental applications of these methods range from small-size systems [71–74], industrial processes [59, 75–77] and safety-critical processes such as nuclear and aerospace industries [78–81]. Many nonlinear extensions of PCA and PLS models were developed in the last decades [82–84]. These nonlinear techniques utilize polynomials, splines, neural networks, etc., to build the latent variable relations to the original measurement data. Where iterative solution methods are involved. The kernel methods are first introduced and developed by Scholkopf et al. in [85], the kernel techniques use linear computation methods to extract the latent variables instead of iterative methods, thus they have been more attractive recently in process monitoring.

2.3.1 Fault detection performance evaluation

The performance of any detection algorithm is evaluated in terms of many indicators the most common indicators are:

- **False Alarm Rate (FAR)(%):** percentage of the healthy samples that are detected as faulty.
$$FAR = \frac{Violated\_data}{healthy\_data} \times 100\% \tag{2.3}$$
- **Missed Detection Rate (MDR)(%):** percentage of faulty samples that are not detected.
$$MDR = \frac{Missed\_detection}{faulty\_data} \times 100\% \tag{2.4}$$
- **Detection Time Delay (DTD):** difference time between detection and occurrence of a fault.
$$DTD = T_d - T_f \tag{2.5}$$

where T_d is the detection time and T_f is the fault occurrence time.

2.4 Conclusion

In this chapter, the fundamental theory of fault detection and diagnosis has been provided. The various methods used to detect faults have been described. The two main classes of fault detection techniques are model-based and data-driven. Model-based fault detection and diagnosis methods that depend on the mathematical model of the process are introduced. model-based fault detection use observer-based, parity space approach, interval approach to detect any abnormalities among the process. data-driven methods highly depend on the historical data of the process with they use statistical and machine learning techniques in order to build a comprehensive model of the process for fault detection and diagnosis.

3

PCA and KPCA for fault detection

3.1 Introduction

This chapter presents the main concepts of PCA and the mathematical development PCA model to built fault detection scheme. In addition, KPCA is introduced as nonlinear extension of PCA to monitor nonlinear processes. Finally, we present the drawbacks of KPCA which suffers from high time and space computations.

3.2 Principal component analysis (PCA)

PCA is one of the most well-known multivariate statistical modeling techniques and is widely used in various disciplines, such as in data compression, face recognition, filtering, image analysis, and fault detection [86–90]. PCA can be extremely useful in quality control applications because it allows one to transform a set of correlated variables to a new set of uncorrelated variables that may be easier to monitor with control charts. In PCA, the measurements of m dimension space (where m is the number of observed variables) are linearly orthogonal projected onto a lower-dimensional space (principal component space of dimension $l < m$) by maximizing the variances of the projections. Many versions of PCA have also been developed, which include recursive PCA (RPCA) [89], multiscale PCA (MSPCA) [91], moving window PCA (MWPCA) [92], multiway PCA [93], dynamic PCA (DPCA) [94], and nonlinear PCA (NLPCA) [95, 96].

3.2.1 Modeling using PCA

PCA is the most utilized multivariate statistical technique, that project the original variables into a new set of orthogonal variables, so that the first components that have the largest variance contain most information. Let $X \in \mathbb{R}^{n \times m}$ be a normalized dataset with m variables and n observations.

$$X = [x_1,\ x_2, ..., x_n]^T = \begin{bmatrix} x_{1,1} & ... & x_{1,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{3.1}$$

PCA can be performed through the eigenvalue decomposition of the covariance matrix of

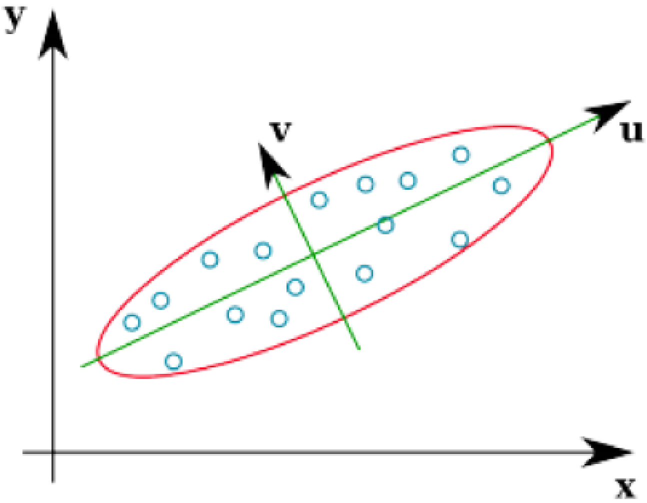


Fig. 3.1 Illustration of PCA in 2D

X. First, the data matrix X is normalized to zero mean and unit variance, then the eigenvalue decomposition is performed.

The covariance matrix is given as

$$C = \frac{1}{n-1} X^T X \tag{3.2}$$

The covariance as represented in equation 3.2 is an unbiased estimator [11]. While the biased estimator for the covariance is given by:

$$C = \frac{1}{n} X^T X \tag{3.3}$$

The covariance matrix of X can be decomposed as

$$C = P\Lambda P^T \tag{3.4}$$

Where the matrix $P \in \mathbb{R}^{m\times m}$ is the principal component loading vectors. While,

$\Lambda \in \mathbb{R}^{m \times m}$ is a diagonal matrix of the eigenvalues. The data matrix is given as

$$X = TP^T \tag{3.5}$$

where T is called the score matrix, and it is given by

$$T = XP \tag{3.6}$$

The matrix P can be decomposed into

$$P = \left[\hat{P} | \tilde{P}\right] \tag{3.7}$$

where $\hat{P} \in \mathbb{R}^{m \times \ell}$ contains the principal components, $\tilde{P} \in \mathbb{R}^{m \times (m-\ell)}$ is the residual components matrix and ℓ represents the number of retained principal components.

3.2.1.1 Principal components number selection

The number of retained principal components is still a matter of study, many methods have been used to select the appropriate number of PCs.

3.2.1.1.1 Kaiser criteria Guttman-Kaiser criteria is introduced in 1954, and is still the most used method to select the number of components in PCA. Kaiser criteria, also known as k1, collects the components corresponding to those eigenvalues greater than 1. $k1$ method first leads to good results. While other studies show that the $k1$ overestimates the number of retained PCs. Furthermore, selecting components with singular values larger than 1 is discussable, due to the fact singular value equals 1.01 is considered as significant and informative while 0.99 component is not significant [97]. Another major problem was reported by many studies, k1 always retains between $\frac{1}{3}$ and $\frac{1}{5}$ or $\frac{1}{6}$ of the total components.

3.2.1.1.2 Scree Plot The scree test was first proposed in [98] as a method that consists of plotting the eigenvalues of the covariance/correlation matrix in decreasing order then exploring the resulting graph to determine the point where the last drop takes place and the graph starts to be smooth. The reasoning behind this approach can be seen as if the elbow point is dividing the important (major) components from the insignificant (minor) components. This test is simple to apply, but it may be hard to interpret due to the fact that a graphical method without any systematic rule may turns out to be highly subjective. Furthermore, the graph itself might be misleading due to the ambiguity of the elbow due to the gradual sloop of the graph or the existence of more than one elbow [11, 99, 100].

Usually, a cumulative eigenvalue proportion graph is used in parallel with the scree test in order to increase the confidence in the selection made by the method. The test can be carried out in terms of the logarithmic eigenvalue test (LEV), this test does extend the scree test by plotting the logarithms of the eigenvalues $log(\lambda_i) vs i$ instead of the eigenvalues. This approach can increase the interpretability of the plot. In a comparative study carried in [99], 90% of scree test estimation errors were found to be underestimates.

3.2.1.1.3 Cumulative Percent of Variance (CPV) It is well known that the variance is a good measure for the importance of a given principal dimension and for how much information could exist in that dimension compared to the others. Thus, retaining a number of components that corresponds to a certain percentage of the total variance is reasonable. In This dissertation, (CPV) is used. CPV keeps the first ℓ principal components that have a sum of variances is greater than a certain percentage of the total variance(Usually, a percentage between $80 - 85\%$).

$$CPV(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \times 100 \tag{3.8}$$

The matrix T is decomposed into

$$T = \left[\hat{T} \,| \tilde{T} \right] \tag{3.9}$$

$\hat{T} \in \mathbb{R}^{N \times \ell}$ represents the principal scores and $\tilde{T} \in \mathbb{R}^{N \times (m-\ell)}$ is the residual scores.

3.2.2 PCA-based fault detection

The common statistics used to measure the variation in principle and residual subspaces, called: the Hotelling T^2 statistic and Squared Prediction Error (SPE) Q [87].

3.2.2.1 The Hotelling's T^2 statistic

T^2 represents the variability in the principle components subspace.

$$T^2 = \boldsymbol{x} \hat{P} \Lambda^{-1} \hat{P}^T \boldsymbol{x}^T \tag{3.10}$$

Where $\boldsymbol{x}$ is a new data sample vector.

The upper limit that the index T^2 should be below to have the normal operation is defined as

$$T_\alpha^2 = \frac{(N^2 - 1)\, l}{N\,(N - \ell)} F_\alpha\left(\ell, N - \ell\right) \tag{3.11}$$

with α be the significant level and $F_\alpha\left(\ell, N-\ell\right)$ Fisher-Snedecor distribution value corresponding to ℓ and $N-\ell$ degrees of freedom.

3.2.2.2 Squared Prediction Error (SPE)

The Q is the norm of residual space, and it is given as

$$Q = \boldsymbol{x}\tilde{P}\tilde{P}^T\boldsymbol{x} = \|\tilde{\boldsymbol{x}}\| \tag{3.12}$$

The upper control limit of Q is defined as

$$Q_\alpha = \theta_1\left[\frac{z_\alpha h_0\sqrt{2\theta_2}}{\theta_1} + 1 + \frac{h_0\theta_2\left(h_0-1\right)}{\theta_1^2}\right]^{\frac{1}{h_0}} \tag{3.13}$$

Where $\theta_i = \sum_{j=\ell+1}^{m}\lambda_j^i \; i = 1,2,3$, $h_0 = 1 - \frac{2\theta_1\theta_2}{3\theta_3^2}$, and z_α is the value of standard normal distribution corresponding to $(1-\alpha)$ confidence level.

3.2.2.3 Combined Index φ

The combined index is proposed by Yue and Qin in [101], it combines the SPE and T^2 indices. φ is given as:

$$\varphi = \frac{T^2}{T_\alpha^2} + \frac{Q}{Q_\alpha} \tag{3.14}$$

3.2.3 PCA drawback

In PCA, it is assumed that the process variables are linearly correlated. Meanwhile when a process shows nonlinear characteristics, a linear PCA model might not provide proper monitoring performance. In order to address this problem, many nonlinear PCA extensions have been developed. Kramer has proposed a nonlinear PCA based on an auto-associative neural network [82]. while a nonlinear PCA which combines principal curve and neural networks is proposed by Dong et al. in [83]. Other nonlinear PCA techniques have been suggested by Cheng et al. [102]; Hiden et al. [103]; Jia et al. [104]; Kruger et al. [105].

One of the most effective methods is kernel principal component analysis (KPCA), introduced by Schölkopf et al. [16], It maps observations from the original space to a higher dimensional feature space where PCA is performed. KPCA shows a successful monitoring results in process monitoring [15,44,87,106]. The detection of abnormalities using KPCA is performed using the same statistical fault indices T^2, Q, and combined index φ [107].

3.2.4 Limitations of data-driven fault detection

Being heavily based on process data, MSA FD methods performance is deteriorated due to noisy and inaccurate measured data. Usually, collected process data is skewed, corrupted by noise, including some incorrect measurements and errors, incomplete with missing samples, and does not follow an exact distribution [108]. As a direct result, several drawbacks are met in the design and application stages of these methods. The first step in data-driven methods is the choice of an informative set of process variables, carrying relevant information about process behavior and faults symptoms. Data is acquired from all interconnected process stations and unit operations with an appropriate sampling time according to the process dynamics. Collected process data generally undergoes pre-processing and it is used in both statistical learning and parameters tuning while constructing a statistical process model as well as projection and generation of statistics which act as faults indices, faults are here detected through the evaluation of resulting statistics using some control limits (thresholds) defined as parameters in a statistical learning stage. Theoretically speaking most MSPM FD approaches rely on the assumption that the process is operating under ideal conditions, and its collected multivariate data follows an exact distribution. Typically, such conditions do not exist in engineering systems. Data imperfectness and outliers raise the difficulty of extracting the statistical properties of the analyzed signals of a given industrial process making it difficult to define the best random phenomena associated with signals under study. This consequently affects the construction and analysis of (i) statistical model; (ii) calculated parameters including thresholds, and (iii) generated statistics as well as their evaluation. Most of the data-driven FD methods in general and MSA methods, in particular, are based on a statistical model which is trained through collected sets of process data in parallel with its parameter calculations and tuning; Accordingly, imperfectness in training data sets is directly mapped into imperfectness of the trained model and its parameters which are of invaluable importance in all FD and PM stages affecting the whole performance. Moreover, online FD is based on the evaluation of the generated statistics obtained through projections of instantaneous measured data; Data outliers sequentially cause several outliers in the generated statistics acting as the process fault indicators and consequently affect the FD procedure causing false and missed alarms. More importantly, the statistical control limits in all MSA FD methods are calculated based on both the trained model and projected process data of normal operation. It is worth mentioning that these parameters are hence extremely sensitive to data outliers, while they play a crucial role as thresholds in statistics evolution and decision making about the process operation status and its faults. Such severe shortcomings due to data imperfectness either in the design or monitoring stages of FD methods lead to inaccurate statistical information extraction, uncertain parameters tuning, as well as erroneous decision-making problems. Major drawbacks are reflected in the overall FD performance causing extensive amounts of missed and/or false alarms, these deteriorate the monitoring performance and degrade both FD sensitivity and robustness. Besides corrupted data caused shortcomings, a major drawback can be seen in the last but not less important stage of FD methods design; The evaluation of generated statistics is based on a fixed threshold in most FD tools, this establishes a high degree of trade-off between FD robustness and sensitivity. This degree, however, is directly controlled through fixed thresholds calculated based on a given significance level; If the fixed detection threshold level is set too high, there will be fewer false alarms, but no alarms are activated upon an actual fault occurrence, and this will inhibit the detection. If it is reversely set too low, the large number of false alarms will cause unidentified detection.

3.3 Kullback–Leibler divergence(KLD)

KLD, D_{KL} (also called s information divergence or relative entropy), measures the distance between two density distributions [109]. Applications include characterizing the relative (Shannon) entropy in information systems, randomness in continuous time-series, and information gain when comparing statistical models of inference. In contrast to variation of information, it is a distribution-wise asymmetric measure and thus does not qualify as a statistical metric of spread – it also does not satisfy the triangle inequality. In the simple case, relative entropy of 0 indicates that the two distributions in question have identical quantities of information.

Consider two probability distributions P and Q. where, P is considered as the data, the observations, or a measured probability distribution. meanwhile distribution Q is a theory, a model, a description, or an approximation of the probability distribution P. KL divergence can be directly defined as the mean of the log-likelihood ratio and it is the exponent in large deviation theory [110]. The KL divergence is used in many fields of speech and image recognition, such as determining the similarity of two acoustic models [111–113], computing the best match using histogram image models, clustering of models, and optimization by minimizing or maximizing the KL divergence between distributions [114–116].

The divergence satisfies three properties, hereafter referred to as the divergence properties:

1. Self similarity: $D(P \parallel P) = 0$.

2. Self identification: $D(P \parallel Q) = 0$ only if P = Q.

3. Positivity:$D(P \parallel Q) \geq 0$ for all P, Q.

For discrete probability distributions P and Q defined on the same probability space, $\mathcal{X}$, the relative entropy from Q to P is defined to be

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{3.15}$$

which is equivalent to

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log \left(\frac{Q(x)}{P(x)} \right) \tag{3.16}$$

In other words, it is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P. Relative entropy is defined only if for all x, $Q(x) = 0$ implies $P(x) = 0$ (absolute continuity). Whenever $P(x)$ is zero the contribution of the corresponding term is interpreted as zero because $\lim_{x \to 0^+} x \log(x) = 0$. For distributions P and Q of a continuous random variable, relative entropy is defined to be the integral

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \tag{3.17}$$

where p and q denote the probability densities of P and Q.

3.4 Kernel PCA (KPCA)

3.4.1 Kernel trick

The kernel trick seems to be one of the most confusing concepts in statistics and machine learning. It is introduced to deal with linearly inseparable data is to project it onto a higher dimensional space where it becomes linearly separable [117–124]. Let us call this nonlinear mapping function ϕ onto higher dimension space, so that the mapping of a sample x of dimension n can be written as $x \rightarrow \phi(x)$ where $\phi(x)$ of dimension h. Now, the term "kernel" describes a function that calculates the dot product of the images of the samples x under ϕ. An important property of the feature space is that the dot product of two vectors ϕ_i and ϕ_j can be calculated as a function of the corresponding vectors x_i and x_j

$$\phi_i^T \phi_j = k(x_i, x_j) \tag{3.18}$$

The function $k(.,.)$ is called the kernel function, and there exist several types of these functions. Some popular kernel functions are polynomial functions, radial basis (Gaussian) functions, and Sigmoidal functions [107, 125–129]. To illustrate how a nonlinear mapping to an expanded dimensional space can change a nonlinear distribution to a linear distribution, the following illustrative example is given: suppose we have a nonlinear process with two variables, x1 and x2, and there are two data sets; one set has normal measurements and the other one faulty measurements. Figure 3.3 shows the plots of these data sets; the normal measurements are marked with blue dots and the faulty ones with orange dots. In this case, it is impossible to linearly separate the normal data from the faulty one.

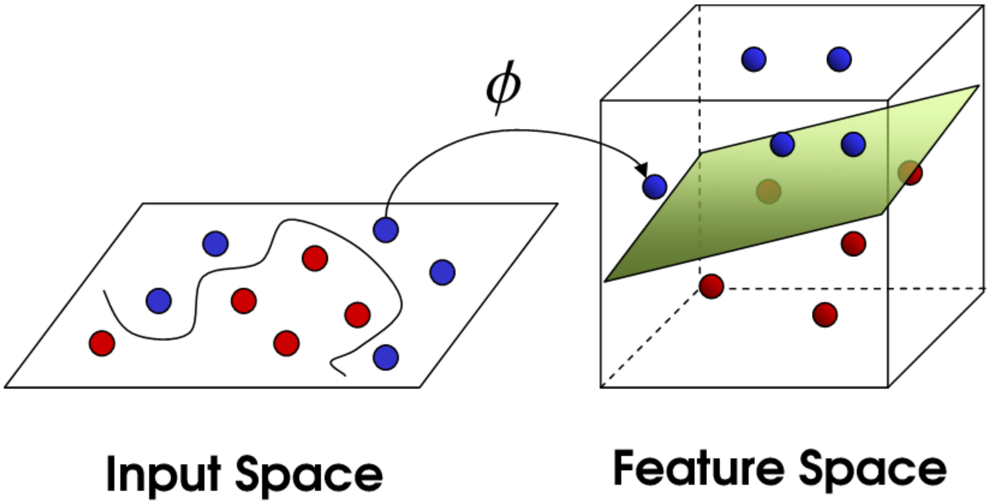


Fig. 3.2 Feature space mapping.

After the following transformation:

$$\phi(X) = \phi \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = i \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \tag{3.19}$$

Figure 3.4 shows the mapped data onto 3D space. it is really easy to separate the normal and faulty measurements with linear PCA. Therefore, even though the original data is nonlinear in a bi-dimensional space, its map to a tri-dimensional space is linear. Furthermore, data becomes linearly separable (by a 2-d plane) in 3-dimensions.

The kernel trick provides a solution to the non-linearly separated data problem. The “trick” is that the kernel method represents the data only through a set of pairwise similarity comparisons between the original data observations x (with the original coordinates in

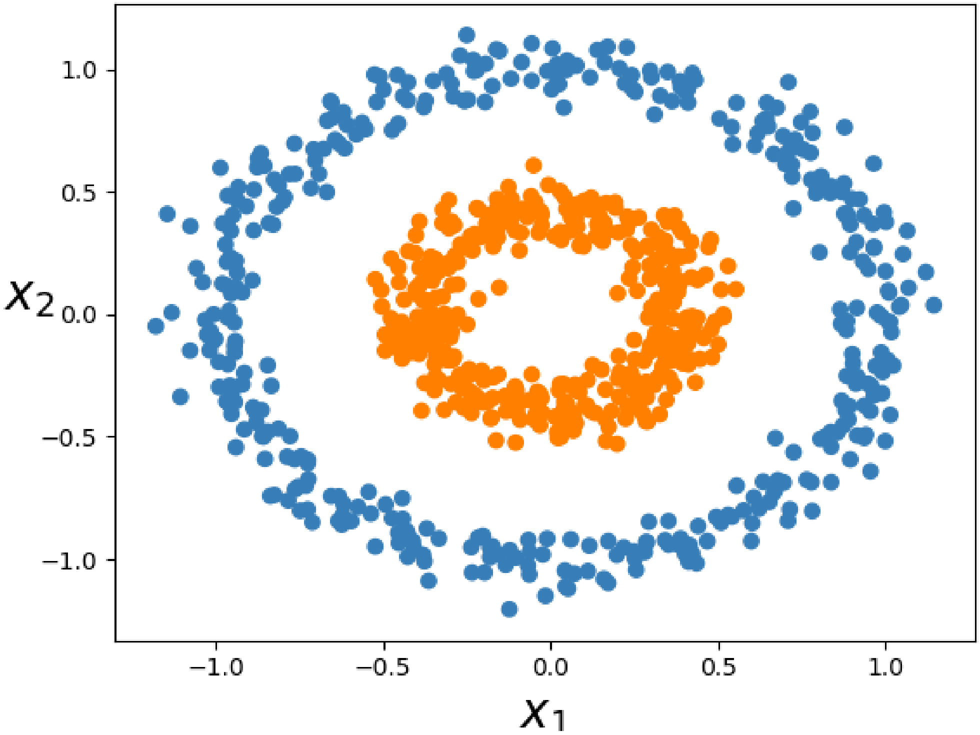


Fig. 3.3 Data in 2D dimension space

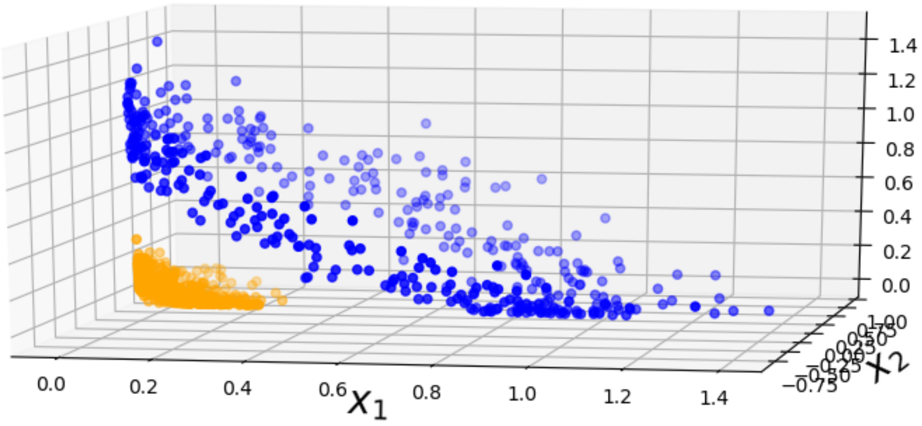


Fig. 3.4 Data mapped into 3D dimension space

the lower dimensional space), instead of explicitly applying the transformations $\phi(x)$ and representing the data by these transformed coordinates in the higher dimensional feature space.

higher-dimensional transformations can allow us to separate data in order to make classification predictions. It seems that in order to train KPCA, operations of the higher dimensional vectors in the transformed feature space. kernel function accepts inputs in the original lower-dimensional space and returns the dot product of the transformed vectors in the higher dimensional space.

3.4.2 Kernel function

There are several kernel functions that are used in KPCA algorithm, the commonly used kernel functions are

- Polynomial Kernel:
$$K(x, y) =< x, y >^d \tag{3.20}$$

- Sigmoid Kernel:
$$K(x, y) = tanh(\beta_0 < x, y > +\beta_1) \tag{3.21}$$

- Radial basis Kernel (gaussian):
$$k(x, y) = exp\left[\frac{-\|x-y\|^2}{2\sigma^2}\right] \tag{3.22}$$

The gaussian, or radial basis, a kernel function is used in this dissertation; The Gaussian kernel function is given as

$$k(x_i, x_j) = exp\left[-\frac{(x_i - x_j)^T(x_i - x_j)}{2\sigma^2}\right] \tag{3.23}$$

where σ is called the kernel parameter or the radius of the kernel function. The value of the kernel parameter affects the fault detection performance. selection of a small value of σ yields kernel function very small or close to 0; while, a very large value of this parameter leads to kernel function very close to 1.

3.4.2.1 Kernel parameter selection

Since the kernel parameter value is a key value in the fault detection scheme. It is very important to select an appropriate value in order to make FDD scheme performs well. Many methods are proposed to compute the value of σ. One of the most recommended techniques is proposed in [130]. This technique aims to select the parameter value as the average minimum distance between two points in the training set.

$$\sigma^2 = c\frac{1}{N}\sum_{i=1}^{N} min_{i\neq j} d^2(x_i, x_j) \tag{3.24}$$

where c is user-defined parameter, $d^2(.,.)$ is the squared distance between two training observations, and N is the number of observations.

3.4.3 KPCA formalization

KPCA main concept is to map the input space onto higher dimension feature space $\mathcal{H}$ through a nonlinear function Φ where it can easily extract features.

$$\Phi\ : x_i\ \in\ \mathbb{R}^m\ \ \rightarrow\ \Phi\left(x_i\right)\ \in\ \mathbb{R}^h \tag{3.25}$$

The covariance of data in the feature space Σ^F is given as:

$$\Sigma^F = \frac{1}{n}\mathcal{X}^T\mathcal{X} = \frac{1}{n}\sum_{j=1}^{n}\Phi(x_j)\ \Phi(x_j)^T \tag{3.26}$$

Where it is assumed that $\sum_j^n \Phi\left(x_j\right) = 0$. Σ^F can be diagonalized by eigenvalue decomposition as

$$\lambda V = \Sigma^F V \tag{3.27}$$

Where λ represents all the eigenvalues and V denotes all eigenvectors.

Substituting Eq 3.26 in Eq 3.27 yields

$$\begin{aligned} \Sigma^F V &= (\frac{1}{n}\sum_{j=1}^{n}\Phi(x_j)\Phi(x_j)^T)V && (3.28)\\ &= \frac{1}{n}\sum_{j=1}^{n} < \Phi(x_j), V > \Phi(x_j) && (3.29)\end{aligned}$$

All solution of V lie in the span of feature space. So, there exist coefficients $\alpha_i\ i = 1\ldots n$ such that $V = \sum_{i=1}^{n}\alpha_i\Phi(x_i)$ thus

$$\lambda\sum_{i=1}^{n}\alpha_i < \Phi(x_k), \Phi(x_i) >= \frac{1}{n}\sum_{j=1}^{n}\alpha_i < \Phi(x_k), \sum_{j=1}^{n}\Phi(x_j) >< \Phi(x_j), \Phi(x_i) > \tag{3.30}$$

The kernel trick is introduced to make the computation of the inner product in implicitly.

$K_{i,j} =< \Phi(x_i), \Phi(x_j) >$. so Eq 3.30 can be written as

$$\lambda n K\alpha = K^2\alpha \tag{3.31}$$

such that $\alpha = [\alpha_1 \dots \alpha_n]$. and

$$\alpha = \mathcal{X}V \tag{3.32}$$

$$K = \begin{bmatrix} \phi_1^T\phi_1 & ... & \phi_1^T\phi_n \\ \vdots & \ddots & \vdots \\ \phi_n^T\phi_1 & \cdots & \phi_n^T\phi_n \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & ... & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} \tag{3.33}$$

so we have now

$$\lambda n\alpha = K\alpha \tag{3.34}$$

from Eq 3.34 it is clear that α and λ are an eigenvector and eigenvalue of K, respectively. We multiply eq.3.32 by $\mathcal{X}^T$To solve V.

$$\mathcal{X}^T\alpha = \mathcal{X}^T\mathcal{X}V = \lambda V \tag{3.35}$$

So

$$V = \lambda^{-1}\mathcal{X}^T\alpha \tag{3.36}$$

Now in order to determine α_i and λ_i we first eigen-decompose Eq3.34. Then we use Eq 3.36 to get v_i

$$\alpha_i^T\alpha_i = v_i^t\mathcal{X}^T\mathcal{X}v_i = v_i\lambda_i v_i \tag{3.37}$$

Therefore, α_i should have a norm of $\sqrt{\lambda_i}$.

3.4.4 KPCA fault detection

For a given new sample x, the same fault indices have been used in PCA to detect abnormalities.

The kernel scores are given by:

$$t_j =< v_j, \Phi(x) >= \sum_{i=1}^{n} \alpha_i^j < \Phi(x_i), \Phi(x) > \tag{3.38}$$

where $j = 1 \dots l$ where l is the number of retained components.

$$\begin{aligned} Q &= ||\Phi(x) - \hat{\Phi}_l(x)||^2 \\ &= \sum_{j=1}^{n} t_j^2 - \sum_{j=1}^{l} t_j^2 \end{aligned} \tag{3.39}$$

The control limit of Q is given as

$$Q_\alpha = g\chi^2_{h,\alpha} \tag{3.40}$$

where $g = \frac{c}{2\mu}$ and $h = \frac{2m^2}{c}$. μ and c represent the mean and variance of Q, respectively. The T^2 is defined as:

$$T^2 = t^T \Lambda^{-1} t \tag{3.41}$$

The control limit of T^2 is defined as:

$$T^2_\alpha = \frac{(n^2 - 1)\, l}{n\,(n - l)} F_\alpha\,(l, n - l) \tag{3.42}$$

with α is a significant level and $F_\alpha\,(l, n - l)$ Fisher-Snedecor distribution value corresponding to l and $n - l$ degrees of freedom.

The combined index φ, which combines the previous indices, is given as:

$$\varphi = \frac{Q}{Q_\alpha} + \frac{T^2}{T^2_\alpha} \tag{3.43}$$

The control limit is given as

$$\varphi_\alpha = \mathfrak{g}\mathcal{X}^2_{h,\alpha} \tag{3.44}$$

where

$$\mathfrak{g} = \frac{\frac{l}{(T^2_\alpha)^2} + \sum_{i=l+1}^{n} \frac{\lambda_i^2}{Q_\alpha^2}}{(n - l)(\frac{l}{T^2_\alpha} + \sum_{i=l+1}^{n} \frac{\lambda_i}{Q_\alpha})} \tag{3.45}$$

and

$$h = \frac{(\frac{l}{T^2_\alpha} + \sum_{i=l+1}^{n} \frac{\lambda_i}{Q_\alpha})^2}{\frac{l}{(T^2_\alpha)^2} + \sum_{i=l+1}^{n} \frac{\lambda_i^2}{Q_\alpha^2}} \tag{3.46}$$

3.4.5 KPCA drawbacks

KPCA is a nonlinear PCA technique that can efficiently compute principal components (PCs) in high-dimensional feature spaces using integral operators and nonlinear kernel functions [16]. Despite the recent reported KPCA-based monitoring applications and good monitoring results, the following problems arise: first, the identification of a KPCA monitoring model requires the storage of the symmetric kernel matrix (computation time increase with the number of samples); second, the fault isolation is a much more difficult problem in nonlinear PCA than in linear PCA [96] and the monitoring model is fixed which may produce false alarms if the process is naturally time-varying. The next chapter proposes new methods to solve the issue of the high time and space complexity of KPCA.

3.5 Conclusion

The conventional PCA is introduced in this chapter, the formalization of PCA and the mathematical description is given, it is shown that PCA works well in linearly correlated variables while it wont perform well in the datasets that show nonlinear characteristics. Then KPCA method is introduced as solution to the nonlinearity issue. Mathematical description, different kernel functions, kernel trick, KPCA based fault detection are illustrated. Finally, it is shown that KPCA struggles with high number of observations, it has high computational time and space compared to PCA and other FDD techniques.

4

Reduced KPCA algorithms

4.1 Introduction

This chapter presented the proposed RKPCA methods to solve the high time and space complexity of ordinary KPCA. Three methods have been presented and discussed to address the issue. The first method is used to reduce the training dataset size based on Euclidean distance between training observations to eliminate similar observations. The second method reduces the correlated observations among the training dataset to remove the correlated observations. The third method aims to get rid of the statistically dependent observations from the original dataset, cosine pair-wise distance is used to determine the dependency among the observations. The three methods build a reduced dataset on which the conventional KPCA is applied. Each of these methods requires a selection of thresholds (similarity, correlation, and independence thresholds) thus loss functions are presented to select the appropriate threshold. Each loss function is presented the monitoring performance versus the threshold of each method.

4.2 Reduced KPCA based Euclidean distances

4.2.1 Similarity and Euclidean distance

Let $X \in \mathbb{R}^{N \times m}$ be the normalized data matrix with N observations and m variables. The Euclidean distance between two different rows of the data matrix X is given by

$$d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \, i = 1, ..., N; j = 1, ..., N; i \neq j \tag{4.1}$$

Consequently, $\frac{N(N-1)}{2}$ pairwise Euclidean distances are obtained for that training data matrix X. So, the further apart the two observations are, the more dissimilar they are and the bigger the distance between them is. The more similar the observations are, the closer they are and the smaller the distance between them is. Thus, the pairwise observations, that are quite similar in term of Euclidean distance, provide the same information and their incorporation in the training data set add an extra computation and memory requirements weight to the KPCA-based monitoring scheme. For this, only one representative and non-redundant observation will be maintained. Accordingly, we obtain a reduced training dataset of size r $(r << N)$ on which KPCA model will be built.

Ideally, zero Euclidean distance means that the concerned pairwise observations are totally similar. In the fact, the absolute similarity is unachieved in a practice world due to the noise and uncertainties in measurement devices. So, it is important to select a range of similarity from zero to a given similarity threshold value β. Hence, all pairs of observations with Euclidean distance within this range will be considered similar. The observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are considered redundant and one of them will be removed from the original training dataset if their corresponding Euclidean distance satisfies:

$$d_{i,j} \leq \beta \tag{4.2}$$

Algorithm 1: RKPCA based Euclidean distance algorithm

```
Offline Phase
        1. Given the input data matrix $X \in \mathbb{R}^{N \times m}$
        2. Normalize the input data matrix $X$
        3. Compute the Euclidean distances $d_{i,j}$ of the normalized data
        4. Fix the redundancy threshold $\beta$ based on the Loss functions presented
in Section 4.6.1
        5. Construct the reduced data matrix $X_{red}$ using Algorithm 2.
        6. Normalize the reduced data matrix $X_{red}$.
        7. Apply KPCA algorithm to $X_{red}$.
        8. Get the control limits of the fault detection indices.
Online Phase
        1. Normalize test data using the mean and standard deviation obtained
           from the reduced dataset in step 6 of the reduced training data
        2. Build The reduced test kernel matrix
        3. Compute $T^2$ , $Q$ and $\varphi$
        4. Compare $T^2$ , $Q$ and $\varphi$ with their corresponding thresholds
        5. Make a decision ( Fault or no Fault)
```

Algorithm 2 represents a pseudo-code of redundant observations removal to reduce the size of the training dataset. The different steps of the proposed RKPCA scheme are illustrated in Fig. 4.1.

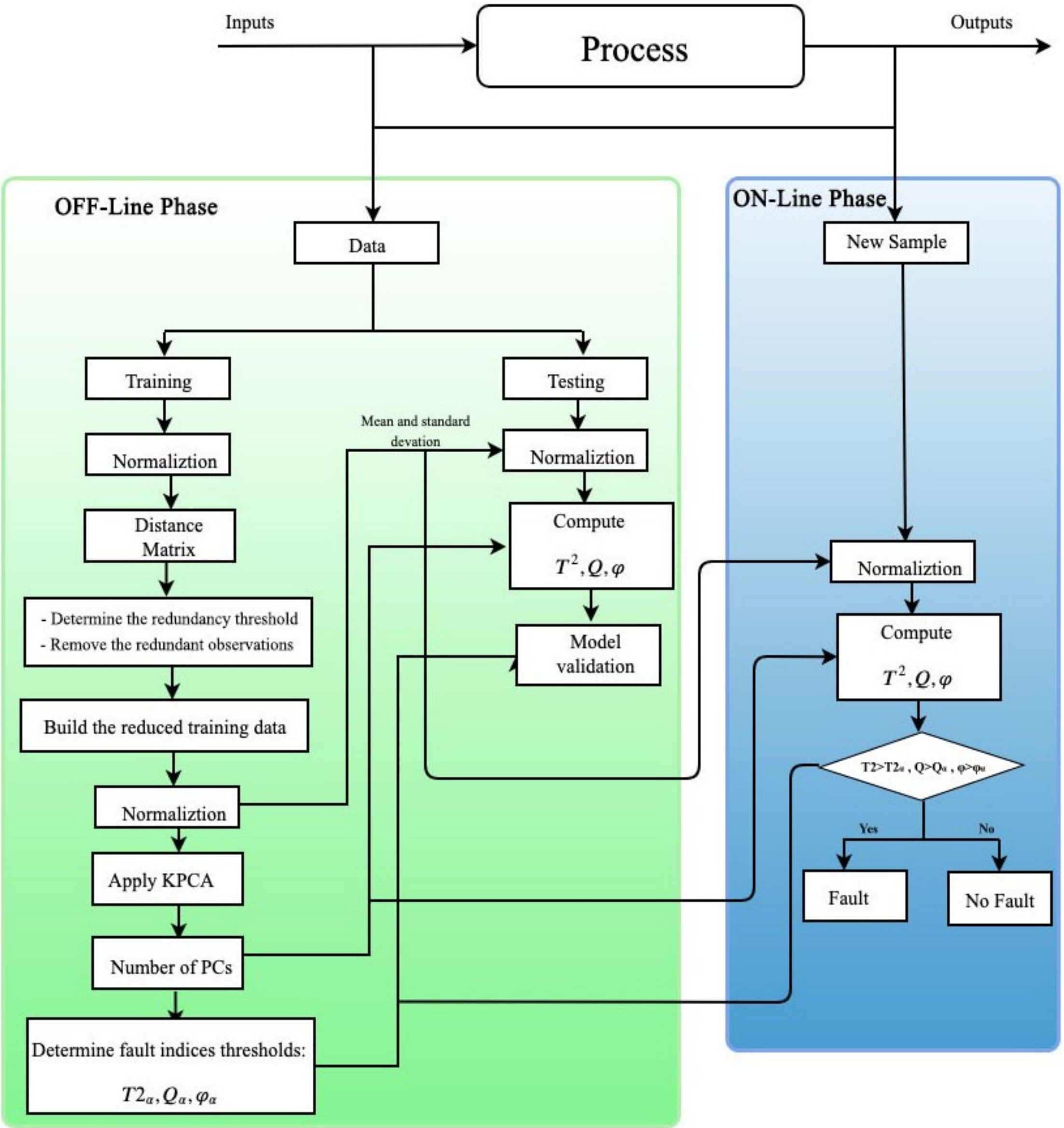

Inputs
Process
Outputs
OFF-Line Phase
ON-Line Phase
Data
New Sample
Training
Testing
Mean and standard devation
Normaliztion
Normaliztion
Distance Matrix
Compute T^2, Q, φ
Normaliztion
- Determine the redundancy threshold
- Remove the redundant observations
Model validation
Compute T^2, Q, φ
Build the reduced training data
T2>T2α , Q>Qα , φ>φα
Normaliztion
Yes
No
Apply KPCA
Fault
No Fault
Number of PCs
Determine fault indices thresholds:
$T2_\alpha, Q_\alpha, \varphi_\alpha$

Fig. 4.1 RKPCA-ED algorithm flowchart

Algorithm 2: RKPCA-ED training dataset reduction

```
Result: $X_{red} \in \mathbf{R}^{r \times m}$
Init. $X \in \mathbf{R}^{N \times m}$ ;
for $i$ = 1 to $N - 1$ do
    for $j$ = $i + 1$ to $N$ do
        $d_{i,j} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2$ ;
        if $d_{i,j} \leq \beta$ then
            Remove observation $\boldsymbol{x}_j$ from $X$;
        end
    end
end
$X_{red} = X$ ;
```

4.3 Reduced KPCA based correlation

Algorithm: The proposed RKPCA-corr

Offline Phase

1. Given the input data matrix $X \in \mathbb{R}^{N \times m}$.
2. Normalize the input data matrix X.
3. Compute the correlation between observations.
4. Select the optimal correlation threshold γ using a loss function.
5. Eliminate observations that have correlation between them greater or equal to γ to build a reduced dataset $X_{red} \in \mathbb{R}^{r \times m}$.
6. Normalize reduced training dataset X_{red}.
7. Build the reduced kernel matrix $K_{red} \in \mathbb{R}^{r \times r}$.
8. Get the control limits of the fault detection indices.

Online Phase

1. Normalize test data using the mean and standard deviation obtained from the reduced dataset size in step6 (offline phase).
2. Build The reduced test kernel matrix.
3. Compute T^2 , Q and φ
4. Compare T^2 , Q and φ with their thresholds.
5. Make decision (Fault or no Fault).

The training data mainly contains highly correlated observations that surely provide the same monitoring information and just add extra computation weight to the training algorithm. Thus, applying KPCA method directly will be a time and space-consuming matter [131].

In this work, it is aimed to reduce the size of a training dataset to overcome the ad-

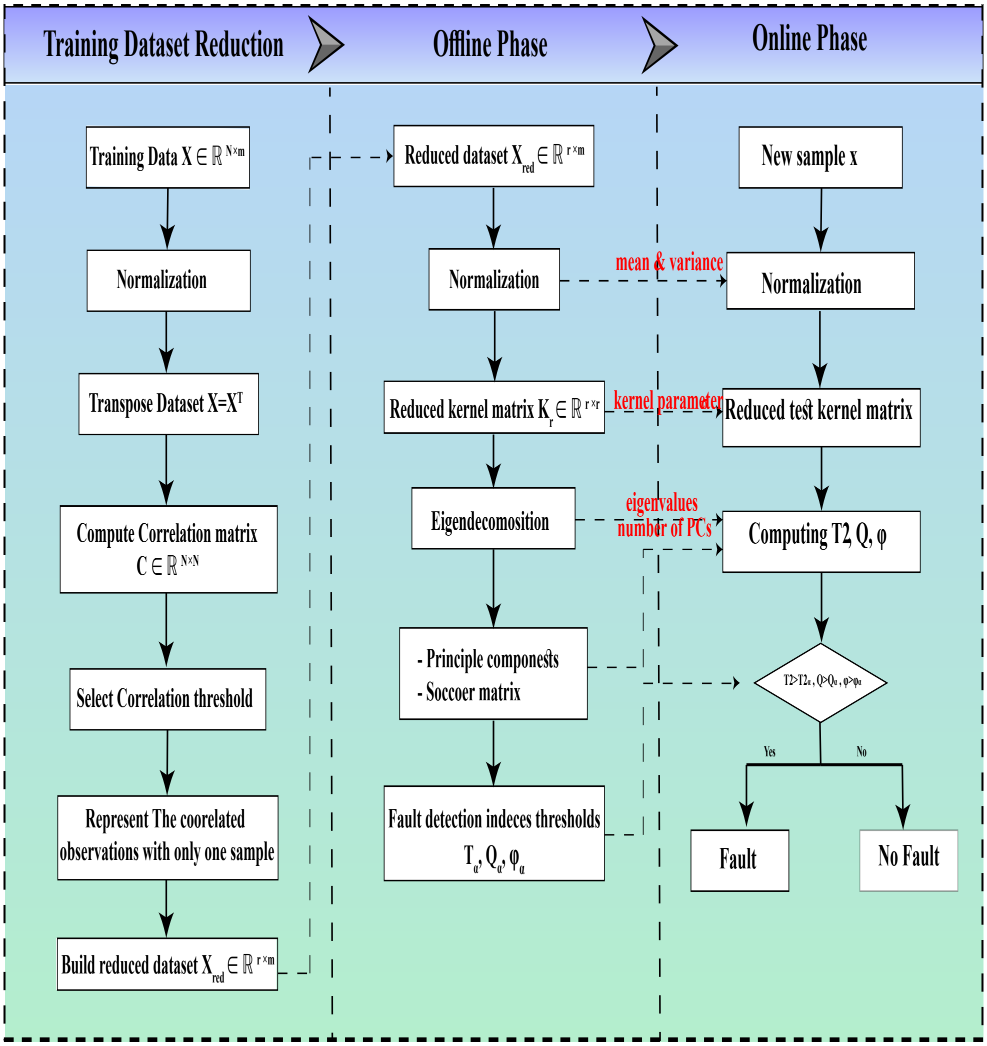

Training Dataset Reduction
Offline Phase
Online Phase
Training Data X ∈ ℝ N×m
Normalization
Transpose Dataset X=X^T
Compute Correlation matrix C ∈ ℝ N×N
Select Correlation threshold
Represent The coorelated observations with only one sample
Build reduced dataset X_red ∈ ℝ r×m
Reduced dataset X_red ∈ ℝ r×m
Normalization
Reduced kernel matrix K_r ∈ ℝ r×r
Eigendecomosition
- Principle componests
- Soccoer matrix
Fault detection indeces thresholds T_α, Q_α, φ_α
New sample x
mean & variance
Normalization
kernel parameter
Reduced test kernel matrix
eigenvalues
number of PCs
Computing T2, Q, φ
T2>T2α , Q>Qα , φ>φα
Yes
No
Fault
No Fault

Fig. 4.2 RKPCA based correlation algorithm flowchart

dressed issues. A new reduced KPCA (RKPCA) algorithm, that can effectively reduce training dataset size and improve the monitoring performance, has been proposed. The idea behind the proposed RKPCA algorithm is to reduce the highly correlated observations and keep only the uncorrelated samples.

RKPCA algorithm consists of three phases. The first phase is the reduction of the dataset size by computing the correlation between each two observations and sorting the correlation coefficients in decreasing order. Given the correlation matrix:

$$\rho = \frac{1}{n-1} X X^T = \begin{bmatrix} cor(x_1, x_1) & \dots & cor(x_1, x_N) \\ \vdots & \ddots & \vdots \\ cor(x_N, x_1) & \cdots & cor(N, N) \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{4.3}$$

where $cor(x_i, x_j)$ represents the correlation between normalized observations x_i and x_j.

The correlation coefficients are now varying from $-1 \leq cor(x_i, x_j) \leq 1$. To avoid the negative sign, the absolute value is taken so $0 \leq cor(x_i, x_j) \leq 1$ high correlated observations have a correlation near to 1. To eliminate the high correlated observations a correlation threshold γ is proposed, so any observations that have $cor(x_i, x_j) \geq \gamma$ should be represented with one sample in the original training dataset building a reduced dataset with a size of r observations where $r << N$. The correlation threshold is selected as one of the previously sorted coefficients based on the loss function presented in section 4.6.2.

The second phase is building RKPCA algorithm using the reduced training dataset to construct the reduced kernel matrix of $r \times r$ size. The reduced kernel matrix consumes less memory and requires less computational time to obtain eigenvalues and eigenvectors in order to determine the fault indices thresholds.

The last phase is the online phase, where any new observation is detected whether it is healthy or faulty. Using the reduced kernel matrix, and the eigenvalues and eigenvector, fault indices for this new sample are computed and compared to the previously computed thresholds so if any index has exceeded its threshold a fault is detected. Figure 4.2 presents the proposed RKPCA technique flowchart.

4.4 Reduced KPCA based cosine

4.4.1 Orthogonality vs Independence

Let X and Y be two random variables, it is said that X and Y are uncorrelated if their covariance is equal to zero.

$$C_{X,Y} = E\{(X - \mu_X)(Y - \mu_Y)\} \tag{4.4}$$

where $E\{.\}$ represents the expectation of a given random variable, $E\{X\} = \mu_X$, and $E\{Y\} = \mu_Y$ The previous equation can be expanded to:

$$C_{X,Y} = E\{XY\} - E\{X\}E\{Y\} = E\{XY\} - \mu_X\mu_Y \tag{4.5}$$

The correlation between random variables X and Y is given as

$$\rho_{X,Y} = \frac{C_{X,Y}}{\sigma_X\sigma_Y} \tag{4.6}$$

where σ_X and σ_Y are the standard deviations of X and Y, respectively. For uncorrelated X and Y means that $C_{X,Y} = 0$, $\rho_{X,Y} = 0$, and $E\{XY\} = E\{X\}E\{Y\}$.

Theorem 4.4.1: Orthogonality

Two random variables are said to be orthogonal if $E\{XY\} = 0$ so if X and Y are uncorrelated, $(X - \mu_X)$ and $(Y - \mu_Y)$ are orthogonal.

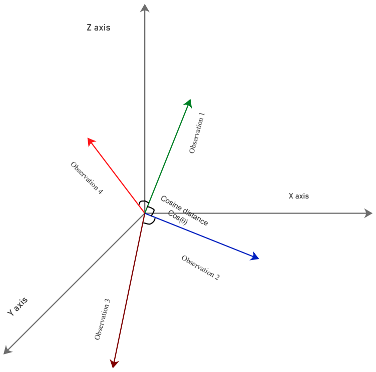


Fig. 4.3 3D plot for the cosine similarity of 4 observations

Theorem 4.4.2: Statistic independence

If two random variables are independent, then they are uncorrelated. Thus any orthogonal random variables are statistically independent.

4.4.2 Cosine similarity

Cosine similarity measures the similarity of two vectors. It computes the cosine of the angle between them. The cosine value is between -1 to 1 for an angle $-\pi \leq \theta \leq \pi$. It judges the orientation of two vectors regardless of their magnitudes. The cosine similarity is equal to zero for orthogonal vectors.

$$cos(\theta) = \frac{< X \cdot Y >}{||X||||Y||} \tag{4.7}$$

Two independent random variable vectors are said to be orthogonal if they have $90°$ angle between them. Thus the cosine similarity is used to determine the statistical dependency between variables. Figure. 4.3 displays an example of the cosine similarity of 4 observations in 3D space. **Cosine pairwise distance** γ equals to one minus cosine similar-

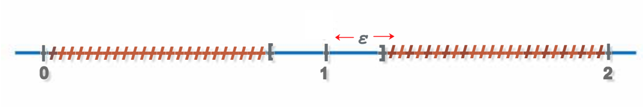


Fig. 4.4 Independence interval

ity. It is used as a metric to determine how similar two observations are, based on the angle between them. The values of cosine pairwise are $0 \leq \gamma \leq 2$.

4.4.3 Dataset reduction

The main drawback of the conventional KPCA algorithm is the high computational complexity that extremely increases with the number of observations. It requires a space complexity of $O(N^2)$ to build the kernel matrix and time complexity of $O(N^3)$ to eigendecompose that kernel matrix. This is a time-consuming matter especially with the modern complex industrial processes that demand a large number of training observations. The aim of this work is to propose a novel RKPCA algorithm that is able to solve the high computation complexity of conventional KPCA techniques and providing a good. The collection of industrial datasets is always vulnerable to noise and uncertainties that affect the quality of that data which leads to slow down KPCA algorithm and deteriorates the monitoring performance and affects the production quality and the safety of the process.

The proposed RKPCA will extract the statistically independent observations and eliminates the statistically dependent observations that surely provide the same information and just rises the computation weight of the training algorithm in order to build a reduced dataset to reserve the same quality as the original dataset to develop an appropriate KPCA.

Cosine pairwise distance is used to determine the independent observations of the training dataset.

Given the cosine pairwise matrix:

$$\Gamma = \begin{bmatrix} \gamma(x_1, x_1) & ... & \gamma(x_1, x_N) \\ \vdots & \ddots & \vdots \\ \gamma(x_N, x_1) & \cdots & \gamma(x_N, x_N) \end{bmatrix} \in \mathbb{R}^{x_N \times x_N} \tag{4.8}$$

where $\gamma(x_i, x_j) = 1 - cos(\theta_{x_i,x_j})$ is the cosine pairwise of x_i and x_j, that have angle of θ_{x_i,x_j} between them.

Thus, any two observations that have a cosine pairwise distance equal to 1 are independent. and they will be kept as training observations while the dependent ones will be eliminated. However, due to the high noise and measurement uncertainties, the independent observation will not have exactly 90° angle between them so the cosine pairwise distance will not exactly equal to 1. To solve that issue, A interval of independence ϵ should be taken around 1 to included the noisy independent samples. So any observation x_i and x_j that satisfy $1 - \epsilon \leq \gamma(x_i, x_j) \leq 1 + \epsilon$ are considered independent observation. Fig. 4.4 shows an example of independence interval around 1. In the proposed RKPCA, the selection of independence interval ϵ that determines the independent observations is a challenging task. If ϵ is taken to be equal to 0 it would select only observations that have pairwise cosine distance $\gamma = 1$ so it will eliminate independent observations that may be considered dependent due to the uncertainties Thus it leads to ignoring important information. Also taking a large interval will include many dependent observations, as a result, it won't reduce enough observations. Figure 4.5 presents a flowchart of RKPCA algorithm. The selection of an appropriate value of ϵ is based on the monitoring performance. a multi-objective loss

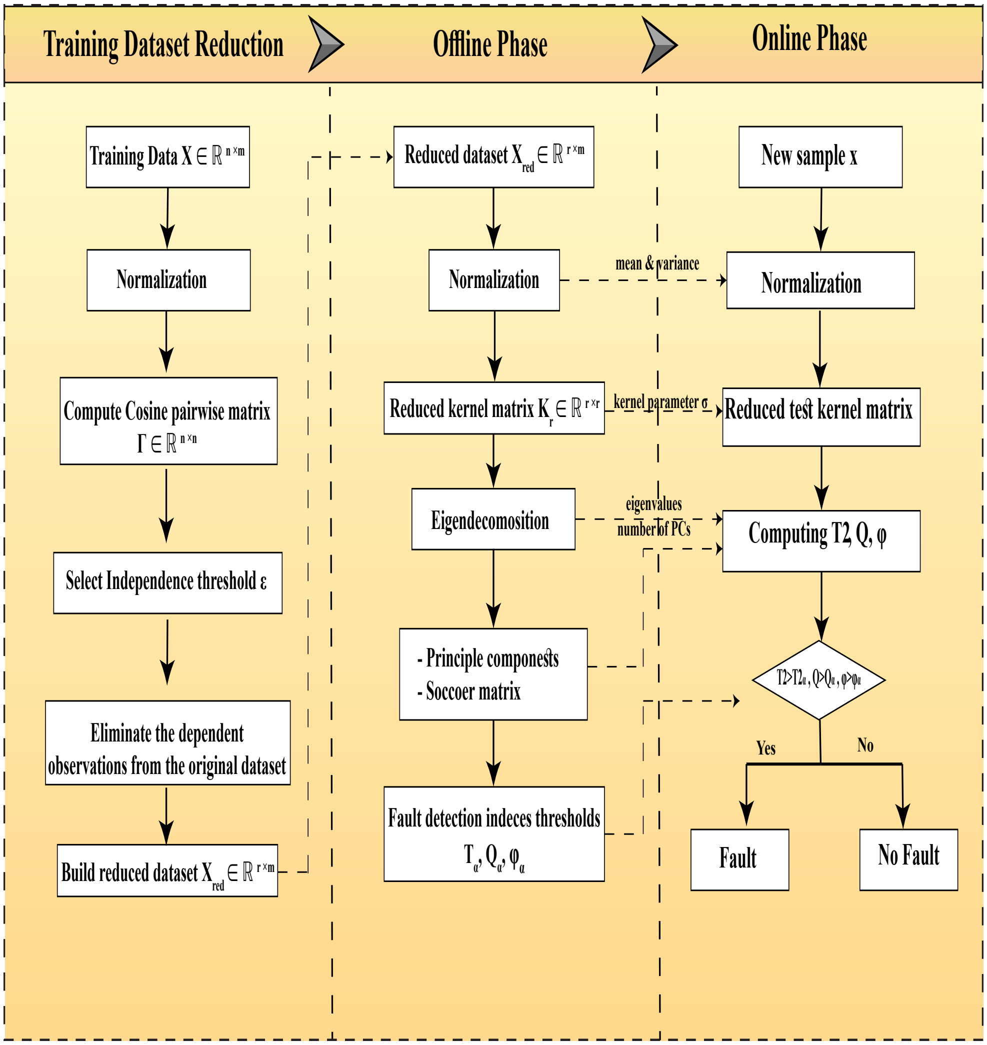

Training Dataset Reduction
Offline Phase
Online Phase
Training Data X ∈ ℝ n×m
Normalization
Compute Cosine pairwise matrix
Γ ∈ ℝ n×n
Select Independence threshold ε
Eliminate the dependent observations from the original dataset
Build reduced dataset X red ∈ ℝ r×m
Reduced dataset X red ∈ ℝ r×m
Normalization
Reduced kernel matrix K r ∈ ℝ r×r
Eigendecomosition
- Principle componests
- Soccoer matrix
Fault detection indeces thresholds
Tα, Qα, φα
mean & variance
kernel parameter σ
eigenvalues
number of PCs
New sample x
Normalization
Reduced test kernel matrix
Computing T2, Q, φ
T2>T2α, Q>Qα, φ>φα
Yes
No
Fault
No Fault

Fig. 4.5 RKPCA Based Cosine algorithm flowchart

function will be used to evaluate the monitoring performance versus ϵ.

Algorithm 3: RKPCA based cosine training dataset reduction

```
Result: $X_{red} \in \mathbf{R}^{r\times m}$
Init. $X \in \mathbf{R}^{N\times m}$ ;
for $i$ = $1$ to $N-1$ do
    for $j$ = $i+1$ to $N$ do
        $\gamma(x_i, x_j) = 1 - cos(\theta_{1,2})$ ;
        if $\gamma(x_i, x_j) \geq 1 \pm \epsilon$ then
            Remove observation $\boldsymbol{x}_j$ from $X$;
        end
    end
end
$X_{red} = X$ ;
```

4.5 Time and space complexity of RKPCA

The analysis of any algorithm should take into consideration the computation time and memory consumption that the algorithm requires to be executed. The developed RKPCA techniques time complexities are summarized in Tables 4.1 and .

The analysis of RKPCA approach time complexity shows that is an algorithm of $O(r^3+N^2)$, where N is the number of original dataset observation and r is reduced dataset number of observations. So reducing training data size to $r < N^{2/3}$ will lead to time complexity of $O(N^2)$ otherwise it will be cubic time complexity $O(r^3)$. In worst case, it is $O(r^3)$, while, in the best case, is $O(r^3)$. the space space complexity is $O(r^2)$ where $r << N$. It is clear that in the worst case the complexity is similar to the ordinary KPCA algorithm but the reduction of the number of observations will improve computational time (the execution

Table 4.1 RKPCA approach time complexity analysis

Method		Cost
RKPCA	**Inti.:** Training data $X \in \mathbb{R}^{N\times m}$	
	Begin:	
	Reduce training data $X_r \in \mathbb{R}^{r\times m}$	N^2
	$K_r = k(x_i, x_j)\|_{i,j=1\ldots r}$	r^2
	eigendecompose $K_r = \hat{P}_r \hat{\Lambda} \hat{P}_r^T$	r^3
	Compute T^2 , Q and φ	r
	Compute T^2_α , Q_α and φ_α	r
	End	
Total	$O(N^2 + r^3 + r^2 + 2r) = O(r^3 + N^2)$	

time taken by the algorithm) and memory space amount required to store the kernel matrix, while in the best case, the time complexity will be a squared. As space complexity, RKPCA comes with $O(N^2)$ and KPCA has $O(r^2)$. It is clear that both algorithms have the same complexity. Although, reducing the number of training observation can improve the storage amount.

4.6 Multi-objective function optimization

In mathematical terms, a multi-objective optimization problem can be formulated as:

$$\min_{x \in X}(f_1(x), f_2(x), \ldots, f_k(x)) \tag{4.9}$$

where the integer $k \geq 2$ is the number of objectives and the set X is the feasible set of decision vectors, which is typically$X \subseteq \mathbb{R}^n$ but it depends on the n-dimensional application domain [132, 133]. Multi-objective design optimization have been implemented in engineering systems in many applications [134–136]. Scalarizing a multi-objective optimization problem is an a priori method, which means formulating a single-objective optimization problem such that optimal solutions to the single-objective optimization problem are Pareto optimal solutions to the multi-objective optimization problem [135, 136]. In addition, it is often required that every Pareto optimal solution can be reached with some parameters of the scalarization. With different parameters for the scalarization, different Pareto optimal solutions are produced. A general formulation for a scalarization of a multiobjective optimization is given as:

$$\min_{x \in X} \sum_{i=1}^{k} w_i f_i(x), \tag{4.10}$$

where the weights of the objectives $w_i > 0$ are the parameters of the scalarization.

The selection of reduced training data set will be based on the minimization of the monitoring performance in terms of FAR, MDR, and DTD. Now, we define our multi objective that sums the three previous fault indices as one function. To select the weighting factors we take the previous studies results and we compute the average FAR as w_1, the average MDR as w_2, and the average DTD as w_3. Also it is recommended to normalize all the indices so DTD is normalized as follows: $DTD_n = \frac{DTD}{faulty\_samples} \times 100$

$$F_1(*) = min(FAR(*)) \tag{4.11}$$

$$F_2(*) = min(MDR(*)) \tag{4.12}$$

$$F_3(*) = min(DTD(*)) \tag{4.13}$$

$$J(*) = min(w_1 F_1(*) + w_2 F_2(*) + w_3 F_3(*)) \tag{4.14}$$

4.6.1 Similarity threshold selection

The selection of the optimal redundancy threshold β that can reduce the training dataset and keep or improve the efficiency of the monitoring algorithm is a challenging issue. For each fault index we compute multi-objective function to obtain the optimal ED that gives the optimal reduced size r. Then we take the summation of the three indices.

$$J_{.}(\beta) = \frac{FAR_{.}(\beta)}{FAR^*_{.}} + \frac{MDR_{.}(\beta)}{MDR^*_{.}} + \frac{DTD_n.(\beta)}{DTD^*_{.}} \tag{4.15}$$

$$J(\beta) \quad = \quad J_{T^2}(\beta) + J_Q(\beta) + J_\varphi(\beta) \tag{4.16}$$

Where $FAR^*_{.}$, $MDR^*_{.}$, $DTD^*_{.}$, represent the desired values for respectively FAR, MDR, DTD of a given statistics. $J_{.}(\beta)$ is the loss function of a selected Euclidean distance threshold for a given fault index. So, the multi-objective optimization problem (4.16), which involves loss functions of the three fault detection indices, is performed to select the optimal threshold β which defines the best trade-off between competing objectives.

4.6.2 Correlation Threshold Selection

similar to similarity threshold, we use the same multi-objective functions to determine the best correlation threshold which gives the optimal reduced size in terms of monitoring performance.

$$\begin{cases} J(T^2,\gamma) = \frac{FAR(T^2,\gamma)}{FT^*} + \frac{MDR(T^2,\gamma)}{MT^*} + \frac{DTD_n(T^2,\gamma)}{DT^*} \\ \\ J(Q,\gamma) = \frac{FAR(Q,\gamma)}{FQ^*} + \frac{MDR(Q,\gamma)}{MQ^*} + \frac{DTD_n(Q,\gamma)}{DQ^*} \\ \\ J(\varphi,\gamma) = \frac{FAR(\varphi,\gamma)}{F\varphi^*} + \frac{MDR(\varphi,\gamma)}{M\varphi^*} + \frac{DTD_n(\varphi,\gamma)}{D\varphi^*} \end{cases} \tag{4.17}$$

$$J(T^2, Q, \varphi, \gamma) = J(T^2,\gamma) + J(Q,\gamma) + J(\varphi,\gamma) \tag{4.18}$$

Where $F\{.\}^*$, $M\{.\}^*$, $D\{.\}^*$, represent the desired values for respectively FAR, MDR,

DTD of a given statistics. $J(\{.\},\gamma)$ is the loss function of a selected correlation threshold for a given fault index. So, the selection of the threshold depends on which index should be optimized.

4.6.3 Independence interval selection

The selection of an optimal independence interval that will be used to reduce the training dataset size in order to enhance the computational time and storage space, and keep or even improve monitoring performance which is a hard task. It is a trade-off between computational cost and monitoring performance.

$$\begin{cases} J(T^2,\epsilon) = \frac{FAR(T^2,\epsilon)}{FT^*} + \frac{MDR(T^2,\epsilon)}{MT^*} + \frac{DTD_n(T^2,\epsilon)}{DT^*} \\ \\ J(Q,\epsilon) = \frac{FAR(Q,\epsilon)}{FQ^*} + \frac{MDR(Q,\epsilon)}{MQ^*} + \frac{DTD_n(Q,\epsilon)}{DQ^*} \\ \\ J(\varphi,\epsilon) = \frac{FAR(\varphi,\epsilon)}{F\varphi^*} + \frac{MDR(\varphi,\epsilon)}{M\varphi^*} + \frac{DTD_n(\varphi,\epsilon)}{D\varphi^*} \end{cases} \tag{4.19}$$

$$J(T^2,Q,\varphi,\epsilon) = J(T^2,\epsilon) + J(Q,\epsilon) + J(\varphi,\epsilon) \tag{4.20}$$

4.7 RKPCA computation time analysis

The evaluation of RKPCA efficiency requires to evaluate the computation time of the algorithm in online phase. We take the original and different reduced training datasets (r=768, r=131, 232, 362, 432) to compute offline phase of RKPCA, then using these reduced training datasets we evaluate the computation time of each fault index of 2000 testing observations. Table 4.2-4.4 lists the computation time of indices T^2, Q, and φ of testing data using original and reduced training data sizes(r=131,232,362,432,768) versus the testing data size N.

Table 4.2 Computation time evaluation of testing data using index T^2

Testing data size	N=1	N=100	N=1500	N=2000
r=131	0.03	1.3	48.8	77.2
r=232	0.05	2.1	56.1	92.2
r=362	0.08	27	68.2	106.4
r=432	0.09	3.6	75.1	114.7
r=768	0.11	4.01	84.8	135.8

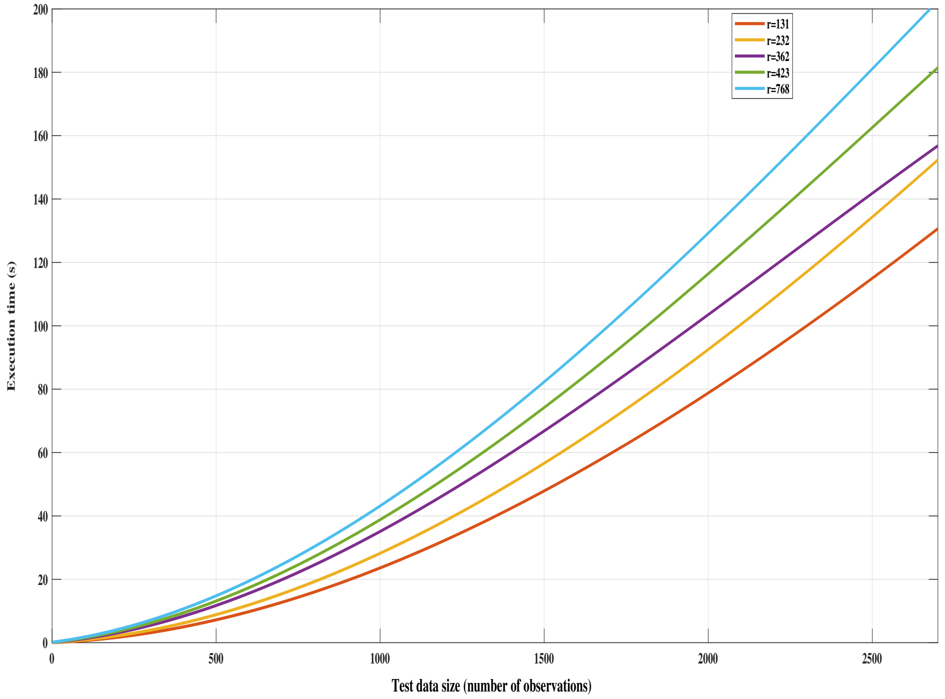


Fig. 4.6 Computation time of RKPCA using T^2 index versus testing data size using different reduced training datasets

Table 4.3 Computation time evaluation of testing data using index Q

Testing data size	N=1	N=100	N=1500	N=2000
r=131	0.04	1.6	49.3	79.6
r=232	0.06	2.5	57.3	94.7
r=362	0.08	3.1	70.1	109.9
r=432	0.10	3.9	76.6	115.6
r=768	0.12	4.03	85.4	132.7

Table 4.4 Computation time evaluation of testing data using index φ

Testing data size	N=1	N=100	N=1500	N=2000
r=131	0.04	1.4	49.0	78.4
r=232	0.06	2.3	56.7	93.9
r=362	0.08	2.9	69.5	108.4
r=432	0.11	3.8	75.8	115.2
r=768	0.11	4.05	82.3	129.6

we already know that the computation time is function of polynomial of degree threes

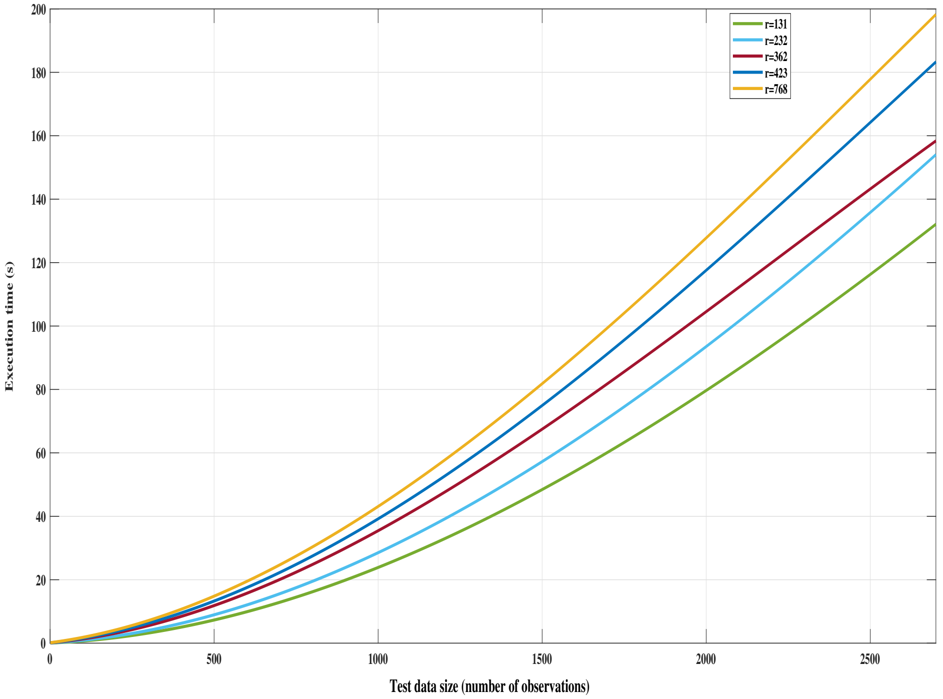


Fig. 4.7 Computation time of RKPCA using Q index versus testing data size using different reduced training datasets

so using regression function we can obtain the computation time function as follows:

$$F_r(n) = \alpha n^3 + \beta n^2 + \gamma r + \delta \tag{4.21}$$

The functions are plotted in figures 4.6-4.8. Using one testing observation yields a computation time inferior than 1 second. These values are less than the sampling time. thus the results of monitoring are obtained before recording new observation. The multi-objective function that is used to obtain the optimal monitoring performance reduces the training dataset to 131 observations. Table 4.5 lists the execution time of different faulty datasets.

Table 4.5 Computation time of different faulty datasets using RKPCA at r=131

	RKPCA		KPCA	
Dataset	Real fault	Simulated faults	Real fault	Simulated faults
T^2	134.3	48.8	283.0	261.0
Q	140.2	49.3	802.1	273.3
φ	138.6	49.0	794.7	269.7

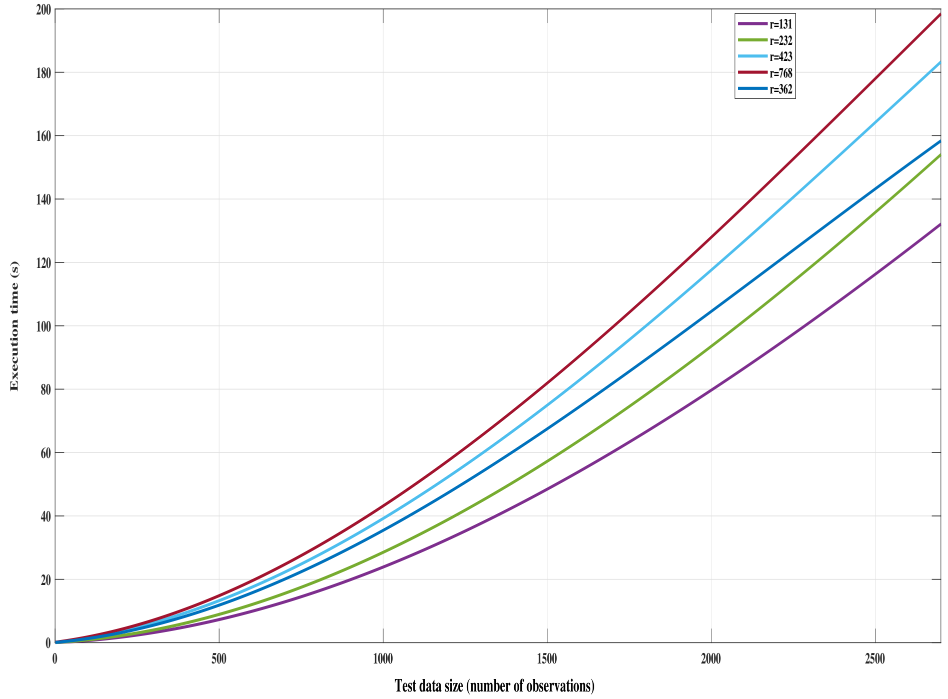


Fig. 4.8 Computation time of RKPCA using φ index versus testing data size using different reduced training datasets

RKPCA algorithm reduced the computation time in real process fault of N=2500 observations to 134.3, 140.2, and 1386 seconds compared to 783.0, 802.1, and 794.7 seconds of KPCA in T^2, Q, and φ respectively. In simulated faults of 1500 observations, computation times are 48.8, 49.3, and 49.0 seconds compared to 2610, 2, and 7947 seconds of KPCA in T^2, Q, and φ respectively. We evaluate the gained computation time which is equal to one minus the ration between computation time of RKPCA and KPCA:

$$G.ET(\%) = (1 - \frac{ET(r)}{ET(r = 768)}) \times 100 \tag{4.22}$$

Where G.ET($\%$) is the gained computation time, ET(r) computation time of RKPCA, and ET(r=768) is computation time. Tables 4.6-4.8 presents the obtained gained computation time results. Reduced dataset of r=131 can gain up to 72.73$\%$ with one testing observations. The gained time decreases until it settles around 40$\%$. The less reduction of training data size the less gained time.

Table 4.6 Gained computation time $\%$ of testing data using index T^2

Testing data size	N=1	N=100	N=1500	N=2000
r=131	72.73	67.58	43.40	43.15
r=232	54.55	47.63	33.84	32.11
r=362	27.27	32.67	19.58	20.48
r=432	18.18	10.22	11.44	15.54

Table 4.7 Gained computation time $\%$of testing data using index Q

Testing data size	N=1	N=100	N=1500	N=2000
r=131	66.67	60.30	42.27	40.02
r=232	50.0	37.970	32.90	28.64
r=362	33.33	28.04	17.92	17.18
r=432	8.33	3.23	10.30	12.89

Table 4.8 Gained computation time $\%$ of testing data using index φ

Testing data size	N=1	N=100	N=1500	N=2000
r=131	66.67	65.43	40.46	39.51
r=232	50.00	43.21	31.11	27.55
r=362	33.33	28.40	15.55	16.36
r=432	8.33	6.17	7.90	10.80

4.8 RKPCA computation space analysis

The storage space depends only on the reduced dataset size r not on the testing data size. we evaluate the storage space using different values of r. After that, using regression function we can get the computation size as function of r. Table 4.9 shows values of the computation space versus r.

Table 4.9 Computation space RKPCA algorithm

r	1	131	768	2000
Storage space(bytes)	5.47×10^2	7.54×10^4	5.49×10^5	1.97×10^6

$$S(r) = \alpha r^2 + \beta r + \gamma \tag{4.23}$$

where $\alpha = 0.2185$ and $\beta = 547.0261$

We plot computation space function in figure 4.9. The optimal monitoring at r=131 requires 7.54×10^4 bytes of memory storage while the full training dataset (768 observations) demands 5.49×10^5 bytes. For any reduced training data size we can now easily obtain its computation space. We evaluate the gained computation time which is equal to one minus

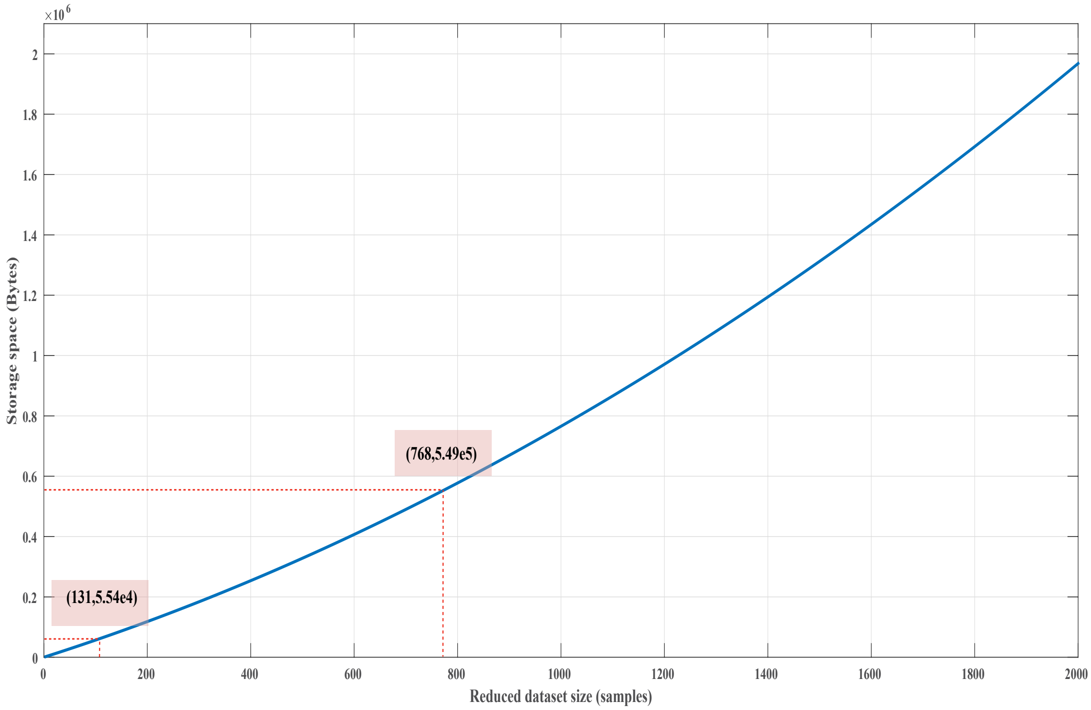


Fig. 4.9 Computation space of RKPCA reduction size

the ration between computation time of RKPCA and KPCA. The gained computation time is given as follows:

$$G.S(\%) = (1 - \frac{S(r)}{S(r=768)}) \times 100 \tag{4.24}$$

Where G.S(%) is the gained storage space, S(r) storage space of RKPCA, and S(r=768) is storage space of KPCA. r=131 gained 86.27% of storage space, r=232 gives 74.86% gained space, and r=362 comes with 58.83% gained storage space. Table 4.10 presents the values of gained computation space.

Table 4.10 Gained computation space RKPCA algorithm

r	1	131	232	362
Gained storage space%	99.90	86.27	74.86	58.83

4.9 Conclusion

In this chapter, the proposed RKPCA methods have been introduced and well explained. These methods have solved the issue of high time and space consumption in KPCA algorithm. the three methods remove the irrelevant training observations that only provide the same monitoring information and adding extra weight to build monitoring algorithms. The proposed methods will be validated in the next chapter where they will be used to monitor industrial processes(Ain El Kebira cement plant rotary kiln process and TEP process), the obtained results will be compared to PCA, KPCA, and recently published RKPCA techniques.

5

Applications, Results, and Discussion

5.1 Introduction

This chapter presents the applications of the proposed methods and the obtained results. RKPCA methods have been applied to the Cement Rotary Kiln of Ain El Kebira. Afterward, they have been applied to the Tennessee Eastman Process (TEP). The obtained results are compared to the conventional PCA, KPCA and recently proposed RKPCA methods(k-means RKPCA [137], reduced Rank-KPCA [129], and RKPCA based PCA [131]).

5.2 Cement rotary kiln process

5.2.1 Process description

The first step in the dry cement production process is to produce flour like raw material by milling limestone, clay, and iron ore mix. This raw material is then fed to the kiln system at the upper end of the preheat tower which is composed of a series of suspending cyclones where heat exchange between feed material and hot gas stream exhausting the rotary kiln is made. Drying, dehydration, and carbon expulsion are initiated wherein. Afterward, in the rotary kiln which is a huge rotating furnace, several chemical reactions occur between calcium and silicon dioxide yielding a new chemical structure called clinker. In the kiln discharge, a cooling system is used to cool hot clinker to preserve its properties using forced air. As a final step clinker and natural gypsum are milled together to get what is commonly known as cement. Ain El Kebira cement plant (first production line) shown in Figure 5.1, where our case study is conducted, is located in the east of Algeria. It has a rotary kiln of $5.4m$ shell diameter and $80m$ length, with 3° incline [10, 138–140]. The kiln can be rotated up to $2.14rpm$ as maximum speed using two $560KWs$ asynchronous motors and produces

Fig. 5.1 Ain El Kebira Cement plant

Table 5.1 Description of different used cement plant process variables

variable	Description	Unit
$x_1,\ x_3,\ x_5, x_7$	cyclones I, II, III, and IV outlets gases pressure for tower I	$mbar$
$x_2,\ x_4,\ x_6,\ x_8$	cyclones I, II, III, and IV outlets gases' Temperature for tower I	℃
x_{10}	cyclones I, II, III, and IV inlet gas's Depression for tower I	$mbar$
$x_{17},\ x_{19},\ x_{21},\ x_{23}$	cyclones I, II, III, and IV outlets gases pressure for tower II	$mbar$
$x_{18},\ x_{20},\ x_{22},\ x_{24}$	cyclones I, II, III, and IV outlets gases Temperature for tower II	℃
$x_{12},\ x_{25}$	the material from tower I and II that entering the kiln Temperature	℃
x_{19}, x_{15}	Tower I and II exhauster fans driving motor power	kW
$x_{11},\ x_{16}$	Tower I and II exhauster fans Speed	$r.p.m$
x_{13}	the smoke filter outlet gas's pressure for tower I and II	$mbar$
$x_{14},\ x_{26}$	the smoke filter outlet gas's Temperature for tower I, and II	℃
x_{27}	The total consumed power two motors rotating the kiln	kW
x_{28}	the excess air from the cooler Temperature	℃
x_{31}	The secondary air temperature	℃
x_{29}, x_{32}, x_{33}	the air under the static grille, repression of fan I, II, and III pressure	$mbar$
x_{30}, x_{34}	the cooling fan I, and fan III respectively Speeds	$r.p.m$
x_{35},x_{37}, x_{39}	pressure of air under the chamber I, II, and III of the dynamic grille, repression of fan IV, fan V, and fan VI respectively	$mbar$
x_{36} ,x_{38}, x_{40}	the cooling fan IV, fan V, and fan VI Speeds respectively	$r.p.m$
x_{41}	the dynamic grille Speed	$strokes/min$
x_{42}	Kiln's head-hood pressure controller output controlling the cooler filter exhaust fans	$r.p.m$
x_{43}	Flow of fuel (natural gas) to the main burner	$Nm3/h$
x_{44}	fuel flow to the secondary burner	$Nm3/h$

clinker with density from 1300 to $1450kg/m^3$ under the normal operation. The plant works with two natural gas burners, the main one placed in the discharge end and the secondary located in the first level of the pre-heater tower without any tertiary air conduct.

The data used to monitor the process is recorded from 44 sensors. The sensors measure temperatures, pressures, speeds, and motors' current. Table. 5.1 illustrates the different

Table 5.2 Cement plant simulated sensors faults description

Fault	Description	Type
SFault1	Fault of 5% magnitude 0 mean and 0.05 std in tower II exhauster fan speed (Sensor 16)	Abrupt
SFault2	Fault of -2% magnitude applied in fuel flow sensor(sensor 44)	Abrupt
SFault3	Faults of $+2\%$ magnitude and slope of 4×10^{-5} incooling fan VI speed sensor (sensors 30)	Linear drift
SFault4	Faults of -2% magnitude and slope of -4×10^{-5} in cooling fan III speed sensor (sensors 34)	Linear drift
SFault5	Fault in temperature and flow sensors(sensors 12, 18, 43) with magnitudes of $[+, -, +2\%]$	Abrupt additive
SFault6	Fault in temperature sensors(4, 6, 8, 12, 24) with magnitudes of $[+, +, +, -, -2\%]$	Linear drift
SFault7	Fault of $+4.5\%$ to -5.5% magnitude in tower I exhauster fan speed (Sensor 11)	Intermittent Abrupt

variables used in the process monitoring. In general, two main datasets are collected from the plant which are used to develop and validate RKPCA model, and afterward to test under different faulty situations the established monitoring scheme. The first dataset is divided into training, testing, and faulty sub-datasets. The training dataset consists of 768 observations (one sample each 20s) which are collected under the healthy operating conditions of the plant for 4 hours and 15 min. It is used to extract a reduced training dataset via the Euclidean distance dissimilarity metric by which a RKPCA model is built. Whilst, the testing dataset contains 11000 samples (one sample each second) which are used to test and validate the developed RKPCA model. The last dataset includes 1500 samples where various simulated sensors faults are carried out to evaluate the monitoring scheme performance [10]. Ain Elkebira cement rotary kiln process has a training dataset of 768 observations most of them are similar, dependent, and correlated. These simulated faults are briefly and adequately described and reported in Table 5.2. On the other hand, the second main dataset represents an actual involuntary process fault and includes 2048 samples.

Table 5.3 RKPCA-ED performance using different selected similarity threshold β

β	X_{red} size	Computation time(s)	J_{T^2}	J_Q	$J\varphi$	J
0.0	768	342	6.23	3.41	7.57	17.21
4.30	237	125	3.59	2.41	2.30	8.30
4.50	165	79	2.94	**2.36**	2.10	7.40
4.67	131	52	**1.98**	2.66	**1.95**	**6.59**
4.80	105	31	2.94	4.37	2.01	9.32

5.2.2 RKPCA based Euclidean distance

5.2.2.1 Training dataset reduction

The original training dataset, before data size reduction, is used to build the conventional PCA and KPCA models for a comparison study with the proposed RKPCA method in the context of faults detection. Subsequently, all Euclidean distances between the original dataset observations are calculated which leads to discarding redundant observations in building the RKPCA model. Thus, at each smaller Euclidean distance, RKPCA model is obtained via a reduced training dataset where 85% of CPV is retained to determine its structure. Monitoring performance is evaluated based on the proposed cost functions outcomes that are fixed to 2%, 2%, and 1 sample as desired FAR, MDR, and DTD, respectively.

Table 5.3 regroups various performance values of the RKPCA model established through the more relevant observations retained from the original training dataset and its corresponding computation time for different similarity thresholds β. It can be clearly seen that the size of the training data is not reduced for $\beta = 0$ due to the noise, uncertainties, and imprecision of the measurement devices. However, the increase of the similarity threshold reduces significantly the training observations, consequently, the associate computation time notably decreases. Nevertheless, large value of β leads to eliminate most of the relevant and non-redundant observations as a result monitoring performances deteriorate accordingly.

To select the optimal similarity threshold β that can determine the redundant observations and gives the optimal monitoring performance, loss functions of the various fault detection indices and their multi-objective metric for different obtained Euclidean distances are depicted in Figure 5.2.(a). It is obvious that the different monitoring indices loss functions have separately various critical similarity values for their corresponding minima. So, from Figure 5.2.(a), the similarity threshold that minimizes the multi-objective problem is $\beta = 4.67$. Consequently, the original training dataset is reduced to 131 observations where 83% of redundant samples have been removed. Besides, the computation time is reduced to 52s.

5.2.2.2 Results and discussion

The obtained similarity threshold provides also the optimal monitoring performance. It is clearly shown in Figure 5.2.(b) that more than 7% of FAR is reduced by the removal of the redundant observation. Furthermore, 0.6% is gained in MDR, whilst one sample increase is recorded in DTD. Figures 5.3 to 5.7 display the monitoring results in the real process and different simulated sensors faults using the various detection indices established via the developed RKPCA technique. These indices are shown before and after the occurrence of faults.

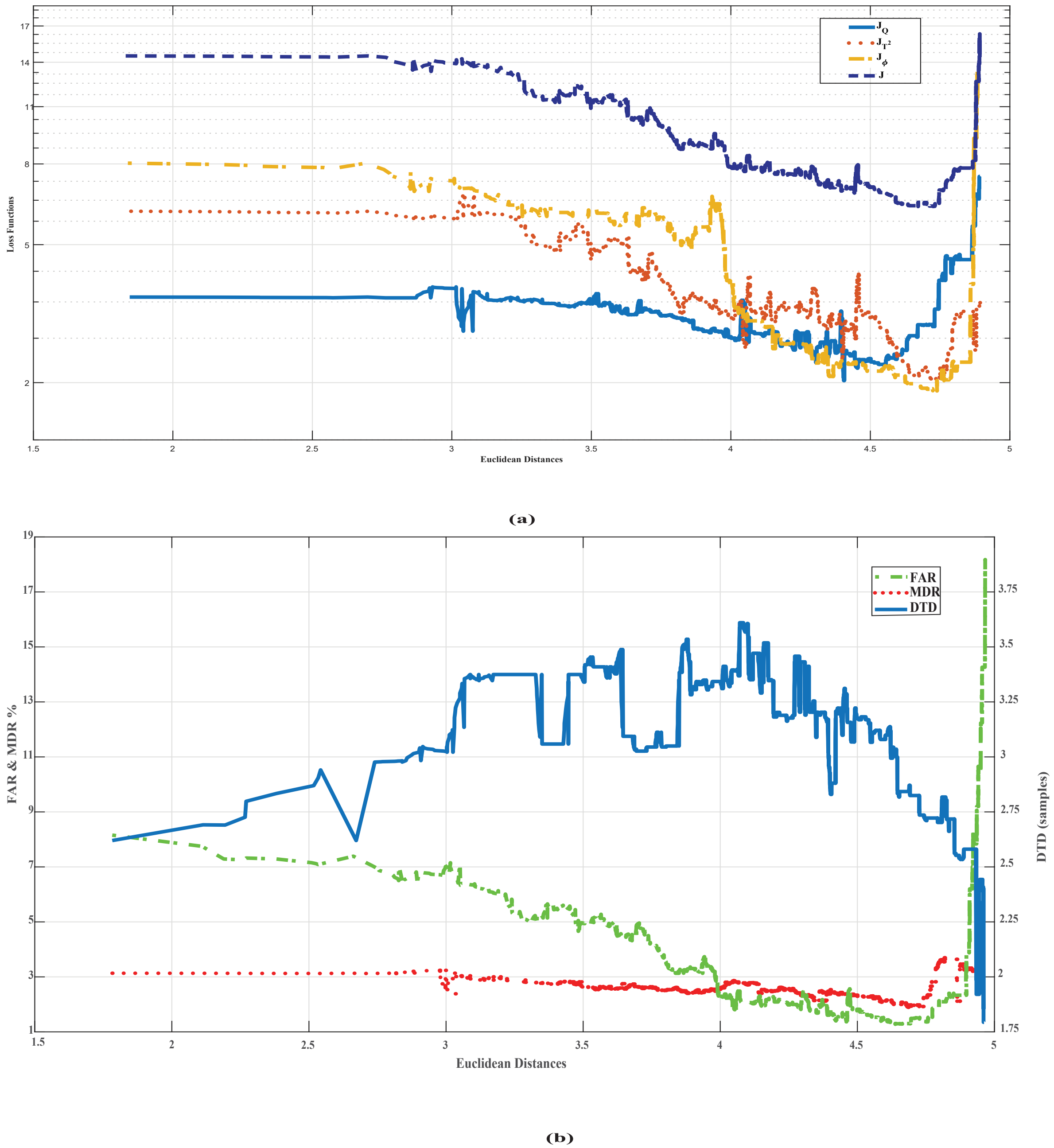
This section shows the obtained monitoring results of Ain El Kebira rotary kiln process using the proposed RKPCA in terms of FAR, MDR, DTD, and the training time required to build the proposed scheme. Also, these results are compared to conventional PCA and KPCA to evaluate the efficiency of the linear and nonlinear characterization of the process and the effect of reducing training dataset on the fault detection and diagnosis performance and solving the high computation and space costs. Besides, the proposed approach is compared to k-means RKPCA [137], reduced Rank-KPCA [129], and RKPCA based PCA [131] methods which focus on the same purpose of reducing the time and space complexity of the conventional KPCA.

Table 5.4 FAR (%) of faults monitoring results

Method		Training	Testing	Real process fault	SFault1	SFault2	SFault3	SFault4	SFault5	SFault6	SFault7
PCA	T^2	2.64	2.40	7.80	0.50	1.00	2.50	0.20	0.50	1.40	0.34
	Q	1.43	1.30	14.70	1.40	1.50	1.10	0.30	1.20	2.10	3.10
	φ	1.04	0.30	2.00	0.00	1.00	1.50	0.00	0.20	0.90	0.00
KPCA	T^2	2.60	2.43	8.01	0.50	1.00	2.50	0.00	0.50	1.40	0.32
	Q	**0.90**	**0.71**	6.68	**0.30**	**1.00**	**0.60**	**0.00**	0.80	**1.60**	**0.88**
	φ	1.04	0.30	2.00	0.00	1.00	0.60	0.20	0.20	0.90	0.00
k-means RKPCA [137]	T^2	1.32	0.63	0.83	0.73	1.02	0.51	1.01	1.13	3.30	0.88
	Q	3.46	3.04	57.23	2.91	2.92	1.36	0.33	3.02	3.20	3.12
	φ	2.90	5.54	21.23	6.80	4.60	0.75	0.52	4.60	6.16	9.93
reduced Rank-KPCA [129]	T^2	1.43	2.20	0.44	0.55	0.60	0.40	0.66	0.88	1.71	0.96
	Q	2.73	1.05	48.99	**0.30**	1.20	0.90	0.20	**0.70**	2.00	4.80
	φ	2.47	0.88	32.51	0.55	1.10	0.70	0.40	0.43	1.90	5.00
RKPCA based PCA [131]	T^2	3.13	3.09	4.98	0.30	0.90	1.20	0.10	0.20	1.10	0.16
	Q	0.52	0.81	6.68	**0.30**	0.40	0.00	0.00	0.00	0.20	0.24
	φ	7.03	6.49	6.50	2.60	2.40	5.50	0.40	1.80	3.40	7.35
RKPCA based ED [10]	T^2	**0.39**	**0.02**	**0.22**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	Q	1.69	3.20	**5.12**	3.50	3.60	1.90	0.50	3.80	2.00	3.79
	φ	**0.13**	**0.02**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

To evaluate and compare the monitoring performance of the well-established RKPCA techniques, the same size of the dataset obtained by the proposed approach (131 samples) is used as the number of selected clusters' centers in the k-means RKPCA method. As well, the reduced Rank-KPCA method has reduced the training dataset to 613 samples. While RKPCA based on PCA technique is able to reduce the training dataset size to 36 observations Table 5.4 reveals the amount of FAR contributed by the three fault indices using PCA, KPCA, different well-established RKPCA, and the proposed RKPCA methods. It can be noticed that from the training and testing datasets, the developed RKPCA method has performed better than the other techniques, particularly in T^2 and φ. Furthermore, it has properly avoided false alarms in all faulty scenarios. On the other hand, FAR recorded by Q is slightly high compared to both conventional PCA techniques, although it is still tolerable. Consequently, the obtained small FAR strongly confirms the accuracy of the proposed RKPCA model and its robustness to monitor the nonlinear process successfully.

Table 5.5 summarizes MDR and DTD results of the three mentioned techniques using T^2, Q, and φ for the real process fault and the different simulated sensors faults. By process expert in the plant, the real process fault occurrence time was estimated to be at sample 450, it can be clearly seen that the proposed RKPCA outperforms all the studied techniques when applied to an actual process fault. Moreover, it shows results relatively equivalent to the conventional RKPCA especially for the DTD and Q index. In general, Q index promptly and correctly detects the almost faults compared to conventional PCA,



(a)

(b)

Fig. 5.2 (a) Different loss functions vs Euclidean distances. (b) Average FAR, MDR, and DTD vs Euclidean distances (RKPCA-ED technique)

KPCA, k-means, and reduced Rank-RKPCA techniques. However, it relatively fails to quickly announce the sensor fault SFault6. While, T^2 and φ indices, obtained using the conventional KPCA, detect with a satisfactory accuracy roughly all the faults than the developed RKPCA method. Even so, MDR is deteriorated for SFault3. As result, it clearly

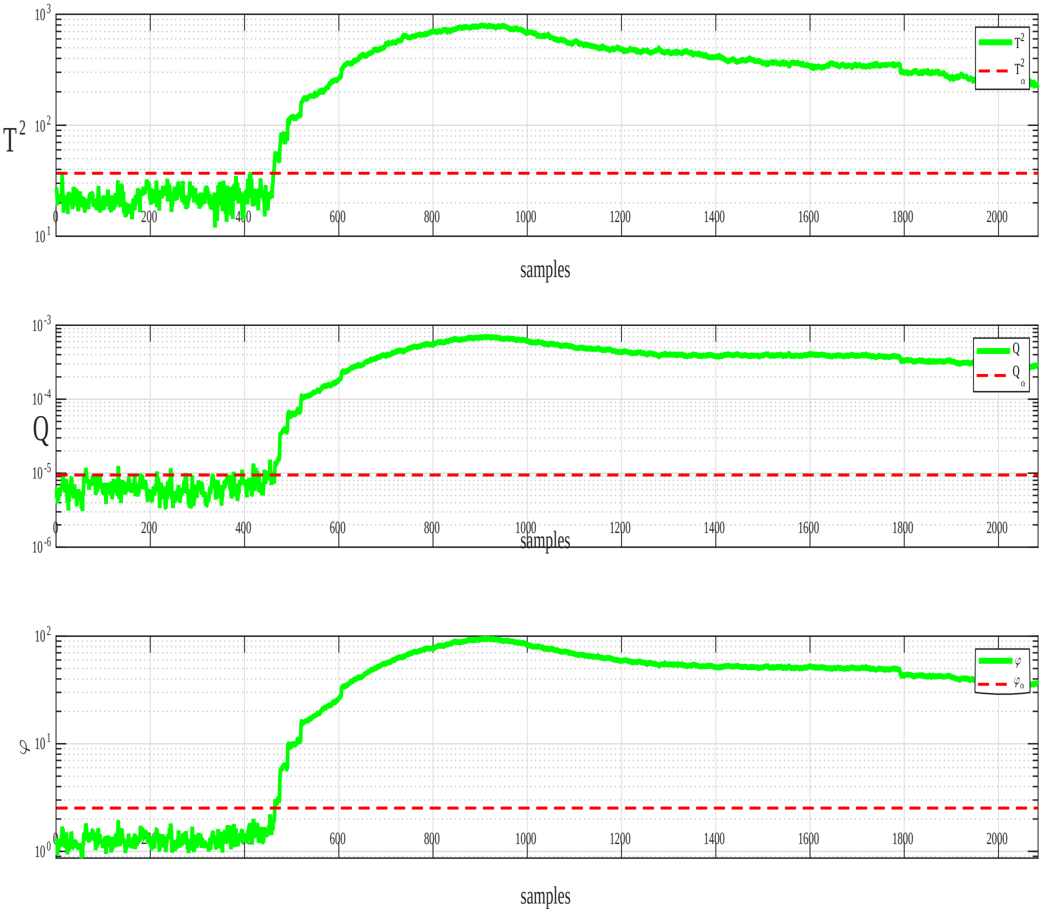


Fig. 5.3 RKPCA-ED monitoring results of a real process fault

appears that the monitoring of abrupt single or multiple faults is easy through the conventional PCA and its nonlinear versions KPCA techniques. Whilst the drift faults, which are considered the most difficult type of faults due to the small development in magnitude over time, are reliably detected with slight delay by the developed RKPCA technique. In spite of the fact that the k-means RKPCA and RKPCA based on PCA techniques take the shortest compute times which are respectively 49 and 27 seconds, the proposed RKPCA technique is still computationally effective which requires 52 seconds to build FDD scheme. It has outperformed the well-established RKPCA methods in terms of FAR using T^2 and φ indices. Furthermore, it has the lowest MDR and DTD using Q index in all faults.

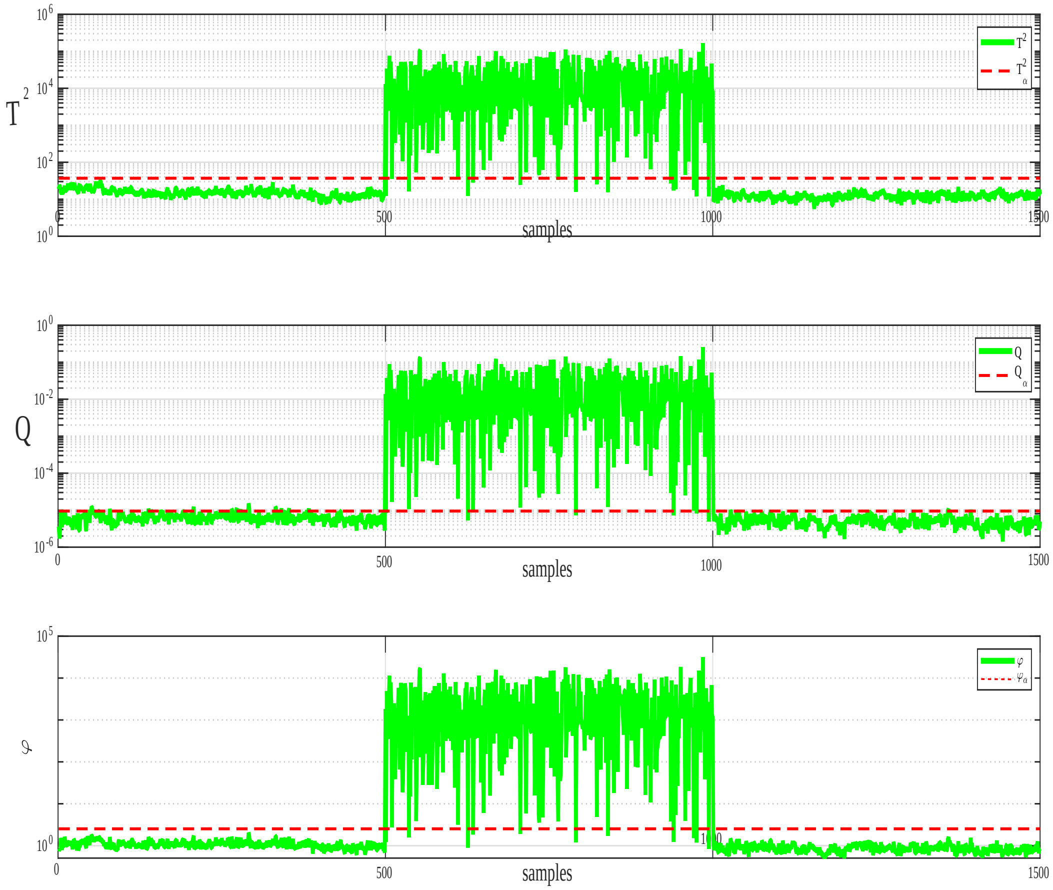


Fig. 5.4 RKPCA-ED monitoring results of the simulated sensor fault SFault1

5.2.3 RKPCA based correlation

5.2.3.1 Training dataset reduction

The training dataset is first zero mean and unit variance normalized. Then, the correlations between samples are computed and sorted in decreasing order. The selection of a correlation threshold that eliminates the correlated observations and builds a reduced dataset is based on one of these sorted correlation coefficients. The first correlation coefficient is selected as correlation threshold, the training dataset is reduced based on that threshold and the monitoring performance is recorded (FAR, MDR, and DTD contributed by the fault indices). Each time the correlation threshold is selected as one of the correlation coefficients to reduce the training observations and record the performance. Loss functions have been used to determine the optimal correlation threshold γ based on the monitoring

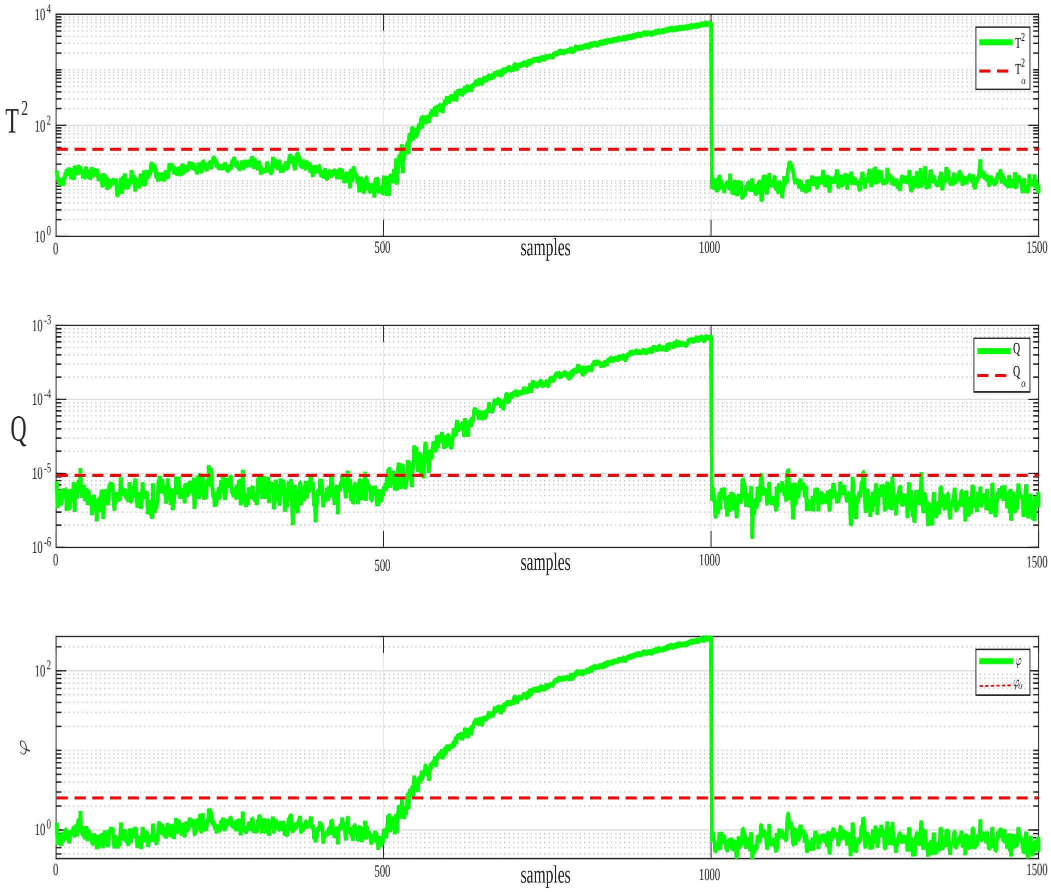


Fig. 5.5 RKPCA-ED monitoring results of the simulated sensor fault SFault3

performance. The performance of the RKPCA is evaluated using loss functions of three fault indices and their average loss function presented in section 4.6.2. At point $\gamma = 0.865$, the average loss function, and index T^2 loss function have their optimal values, while Q and φ get their optimal values at $\gamma = 0.833$ and $\gamma = 0.83$ respectively. Figure 5.8 displays the loss function J and Figure 5.9 shows the loss functions of fault indices T^2, Q, φ plot. It is clear that J and J_{T^2} have optimal values at $\gamma = 0.865$ while J_Q and J_φ optimum at $\gamma = 0.833$ and $\gamma = 0.83$ respectively.

Table 5.6 presents the results of the monitoring performance of RKPCA, computation time and reduced training dataset size (number of retained observations) versus the correlation threshold γ. It can be seen that $\gamma = 1$ will not reduce the training dataset because there is no observations that are entirely linearly correlated. Decreasing the correlation threshold certainly will start accordingly reducing the number of observations. Thus, the execution

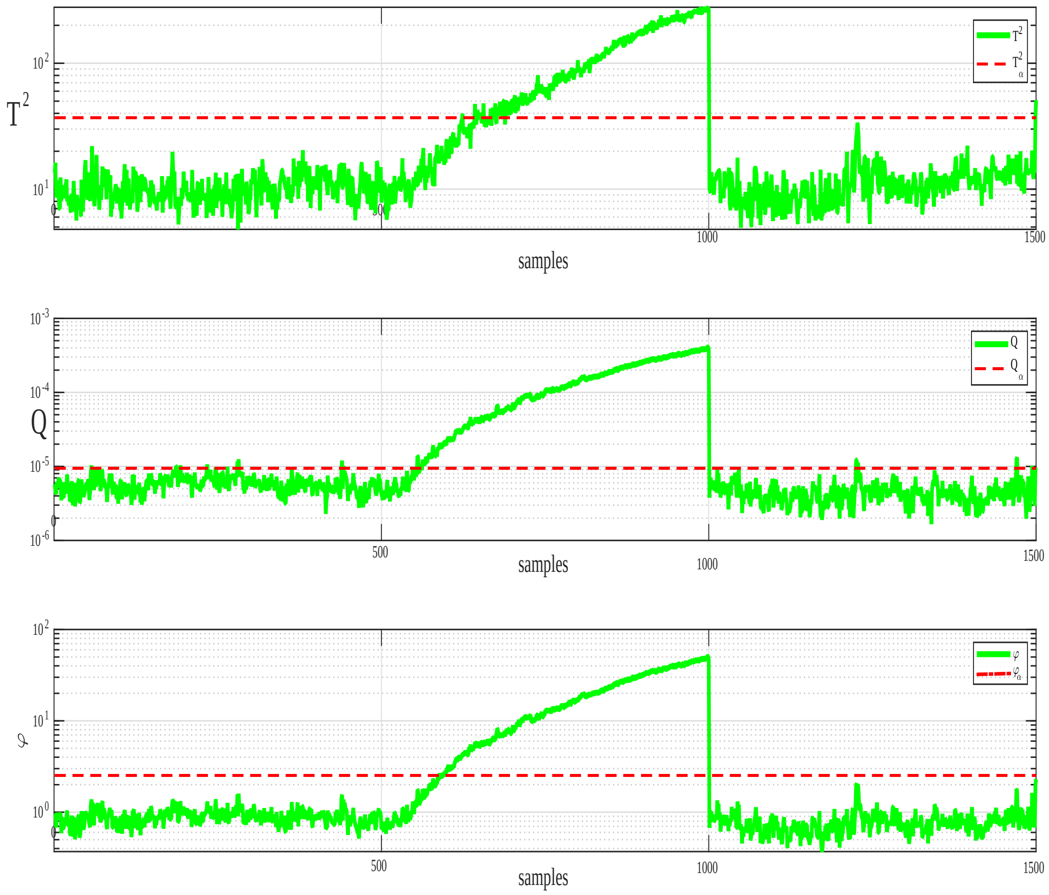


Fig. 5.6 RKPCA-ED monitoring results of the simulated sensor's fault SFault6

time and storage space costs will remarkably decrease. The more decrements in γ more reduction in the dataset size thus more reduction in the execution time, however, it will deteriorate the monitoring performance due to the loss of important information. So the selection of correlation threshold now depends on which index would be improved. In this dissertation, the loss function J that represents the average performance of FAR, MDR, and DTD has been utilized to determine the optimal value of γ. The optimal value is at $\gamma = 0.865$ so any two or more observations that have a correlation between them greater or equal to 0.865 are drooped from the original training data to build a reduced dataset of $r = 513$ observations making a reduction rate of 33.2%.

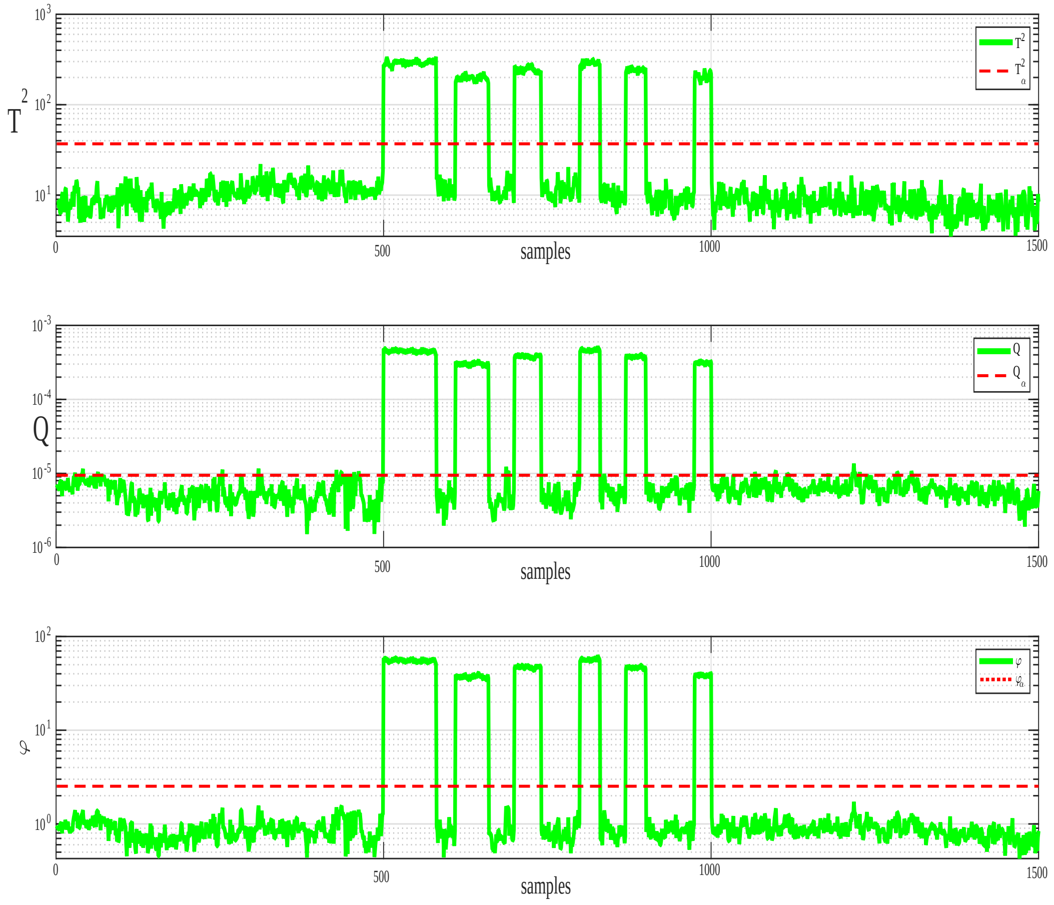


Fig. 5.7 RKPCA-ED monitoring results of the simulated sensor fault SFault7

5.2.3.2 Results and discussion

The reduced dataset of 513 samples is now used to build reduced kernel matrix K_{red} of $r \times r$. K_{red} is eigen-decomposed to generate the eigenvalues and eigenvectors. $CPV\ (l) = 80\ \%$ is used to determine the number of PCs. After retaining l principal components, the fault indices thresholds are computed.

Testing dataset of 11000 observations is used to validate RKPCA scheme. The monitoring scheme should have a few amount of FAR contributed using three fault indices in order to be validated (mainly less than 5%). After the model has been validated, it is used to monitor the different faulty datasets the performance of RKPCA is evaluated in terms of FAR, MDR, DTD. The results are presents in tables 5.7 5.8, 5.9.

Table 5.7 reveals the amount of FAR contributed by the three fault indices T^2, Q,

Table 5.5 MDR (%) and DTD of faults monitoring results

Faults		Real process Fault	SFault1	SFault2	SFault3	SFault4	SFault5	SFault6	SFault7
PCA	T^2	1.98|5	2.40|1	0.00|1	4.70|20	15.60|1	0.00|1	22.50|100	0.00|1
	Q	1.68|4	1.20|1	0.00|1	12.90|7	2.20|3	0.00|1	15.60|56	0.00|1
	φ	1.14|4	1.80|1	0.00|29	7.20|19	6.40|1	0.00|56	19.40|1	0.00|1
KPCA	T^2	0.60|1	2.40|1	0.00|1	7.20|20	15.60|1	0.00|1	22.40|100	0.00|1
	Q	0.60|1	1.20|1	0.00|1	15.90|35	2.80|12	0.00|1	16.80|56	0.00|1
	φ	0.40|1	1.90|1	0.00|1	7.20|29	6.30|19	0.00|1	19.40|56	0.00|1
k-means RKPCA [137]	T^2	0.53|1	6.18|1	0.00|1	8.18|38	35.72|165	0.00|1	54.49|246	0.00|1
	Q	0.70|2	0.79|1	0.00|1	9.60|42	3.99|3	0.00|1	13.97|58	0.00|1
	φ	0.80|2	1.73|1	0.00|1	8.20|36	7.29|33	0.00|1	16.56|56	0.00|1
Reduced Rank-KPCA [129]	T^2	1.88|2	5.18|1	0.00|1	7.18|30	30.53|141	0.00|1	35.13|124	0.00|1
	Q	0.68|2	1.39|1	0.26|2	8.38|78	3.79|5	0.00|1	17.16|56	0.00|1
	φ	0.60|2	2.79|1	0.00|1	7.66|29	6.58|31	0.00|1	19.16|57	0.00|1
RKPCA based on PCA [131]	T^2	0.78|1	4.60|2	0.00|1	7.88|25	20.95|73	0.00|1	28.55|97	0.00|1
	Q	0.62|1	2.23|2	0.00|1	21.15|89	5.38|19	0.00|1	19.76|58	0.00|1
	φ	0.53|1	3.05|2	0.00|1	7.79|28	7.59|3	0.00|1	12.77|58	0.00|1
RKPCA based ED [10]	T^2	0.79|**1**	3.39|**1**	**0.00**|**1**	6.80|29	**14.40**|3	**0.00**|**1**	32.70|**49**	**0.00**|**1**
	Q	**0.48**|**1**	**1.19**|**1**	**0.00**|**1**	**5.90**|**7**	**1.79**|**3**	**0.00**|**1**	**11.80**|125	**0.00**|**1**
	φ	0.91|**1**	2.19|**1**	**0.00**|**1**	7.40|29	**5.20**|19	**0.00**|**1**	18.90|90	**0.00**|**1**

Table 5.6 RKPCA based correlation performance versus γ

γ	Training data size	RKPCA execution time(s)	J_{T^2}	J_Q	$J\varphi$	J
1	768	350	2.457	3.173	2.8368	8.467
0.9	654	303	2.389	3.052	2.743	8.184
0.865	513	240	**2.237**	2.429	2.576	**7.242**
0.833	463	196	3.172	**2.143**	2.0846	7.687
0.83	387	132	3.1876	2.22	**2.051**	8.059
0.8	358	114	3.853	2.504	2.483	8.872
0.78	320	103	3.225	4.493	5.4774	13.632

Table 5.7 FAR contributed by different fault indices

Faults	PCA			KPCA			RKPCA based correlation		
	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ
Training data	2.64	2.33	1.52	2.60	1.25	1.52	**2.10**	**1.20**	**0.86**
Test data	2.55	1.76	1.03	2.60	1.04	0.80	**1.70**	**0.96**	**0.75**
Real process fault	**7.8**	14.7	2.00	8.01	**6.68**	2.00	32.70	10.46	**0.40**
Simulated Fault 1	0.50	1.40	0.00	0.69	**0.37**	0.00	**0.20**	0.80	0.00
Simulated Fault 2	1.00	1.50	1.00	**1.00**	1.00	1.00	1.10	2.40	**0.60**
Simulated Fault 3	2.50	1.10	1.50	2.50	**0.60**	0.60	3.40	**0.60**	**0.00**
Simulated Fault 4	0.20	0.30	0.00	**0.00**	0.00	0.20	0.20	0.10	**0.00**
Simulated Fault 5	0.50	1.20	0.20	0.50	0.80	0.20	**0.20**	**0.50**	**0.00**
Simulated Fault 6	1.40	2.10	0.90	1.40	1.60	0.90	1.90	**0.90**	**0.60**
Simulated Fault 7	0.34	3.10	**0.00**	**0.32**	**0.88**	**0.00**	5.80	**0.88**	0.16
The average	2.04	2.95	0.82	**1.96**	**1.42**	0.72	4.93	1.88	**0.33**

and φ using PCA, KPCA, and the proposed RKPCA-corr. Each row presents the result of the dataset while the last row is an average FAR contributed by each fault index. In the training and test dataset, RKPCA method has contributed less FAR using all the three indices compared to PCA and KPCA. RKPCA method has quietly reduced FAR amount contributed especially using combined index φ to 0.86% and 0.75% in training and test

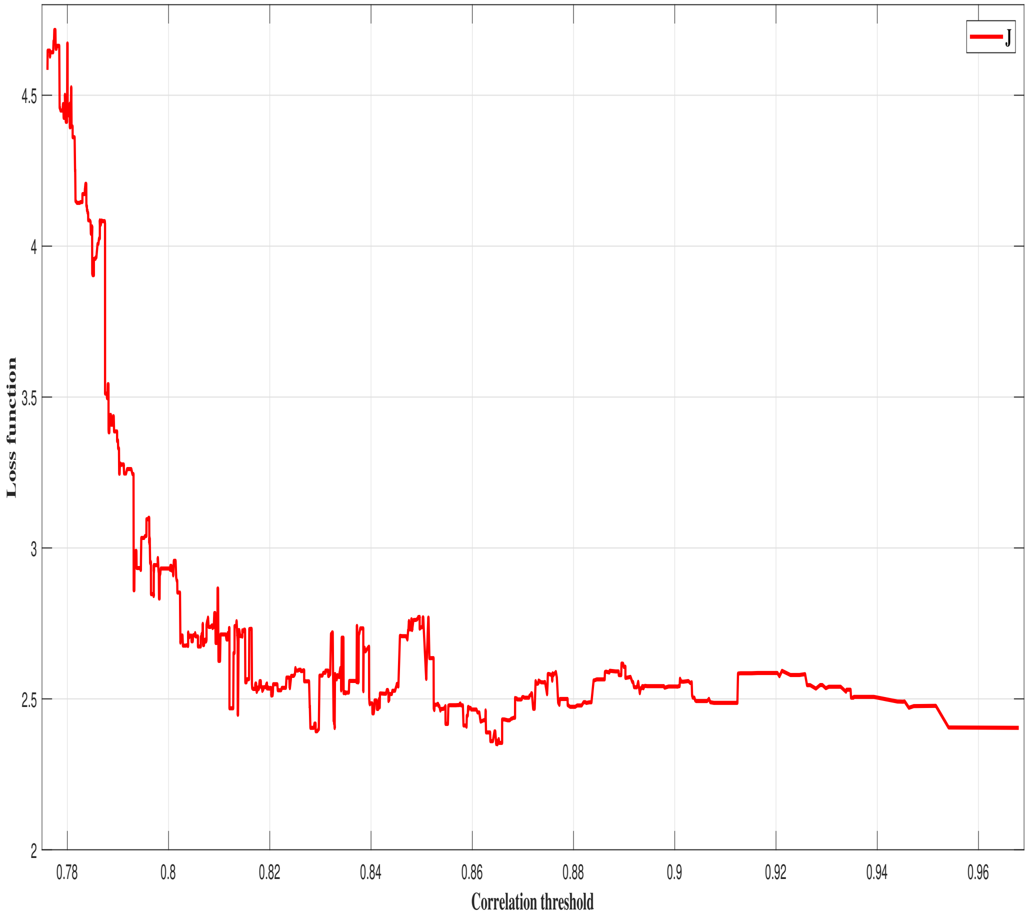


Fig. 5.8 Loss functions J versus γ

Table 5.8 MDR contributed by different fault indices

Faults	PCA			KPCA			RKPCA based correlation		
	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ
Real process fault	1.98	1.68	1.14	0.60	0.60	0.40	**0.00**	**0.55**	**0.30**
Simulated Fault 1	2.4	**1.2**	**1.80**	2.40	**1.20**	1.90	**1.99**	1.59	2.19
Simulated Fault 2	0.00	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Simulated Fault 3	**4.70**	**12.97**	7.20	7.20	15.90	7.20	4.79	13.17	**7.18**
Simulated Fault 4	**15.6**	7.20	6.40	**15.60**	**2.80**	**6.30**	16.96	2.59	6.98
Simulated Fault 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Simulated Fault 6	22.5	**15.60**	**19.40**	22.35	16.76	19.60	**17.36**	16.65	20.55
Simulated Fault 7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
The average	5.90	4.83	4.50	6.01	4.66	**4.25**	**5.13**	**4.31**	4.65

datasets respectively. In the different faulty datasets, RKPCA-corr successfully eliminated FAR in faults 1,3,4,5 while slight false alarms in real process fault and faults 2,6,7 using φ index. meanwhile, the results of T^2 are a little high in real process fault and faults 2,3,4,6, and 7. index Q results good FAR in faults 5,6, and 7. The average FAR results show that

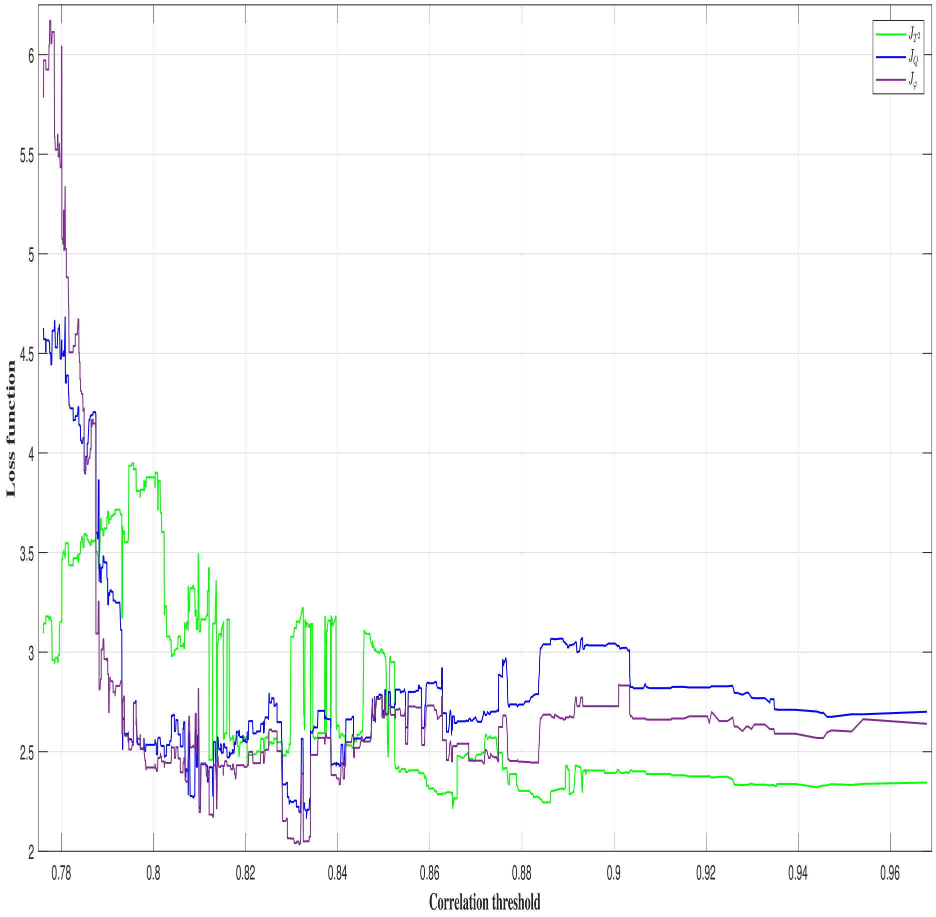


Fig. 5.9 Loss functions of the fault indices versus γ

Table 5.9 DTD contributed by different fault indices

Faults	PCA			KPCA			RKPCA based correlation		
	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ
Real process fault	1	1	1	1	1	1	1	1	1
Simulated Fault 1	1	1	1	1	1	1	1	1	1
Simulated Fault 2	1	1	1	1	1	1	1	1	1
Simulated Fault 3	20	27	29	20	35	29	20	**7**	29
Simulated Fault 4	1	3	19	1	12	19	2	3	34
Simulated Fault 5	1	1	1	1	1	1	1	1	1
Simulated Fault 6	100	56	56	100	56	56	100	56	96
Simulated Fault 7	1	1	1	1	1	1	1	1	1
The average	**15.75**	11.38	**13.63**	**15.75**	13.5	**13.63**	15.87	**8.87**	20.5

our RKPCA contributed the best average using φ while KPCA has the best average in T^2 and Q.

The proposed RKPCA-corr contributes an acceptable FAR rate compared to conventional KPCA, this shows that the reduction of the dataset not only improved the time and

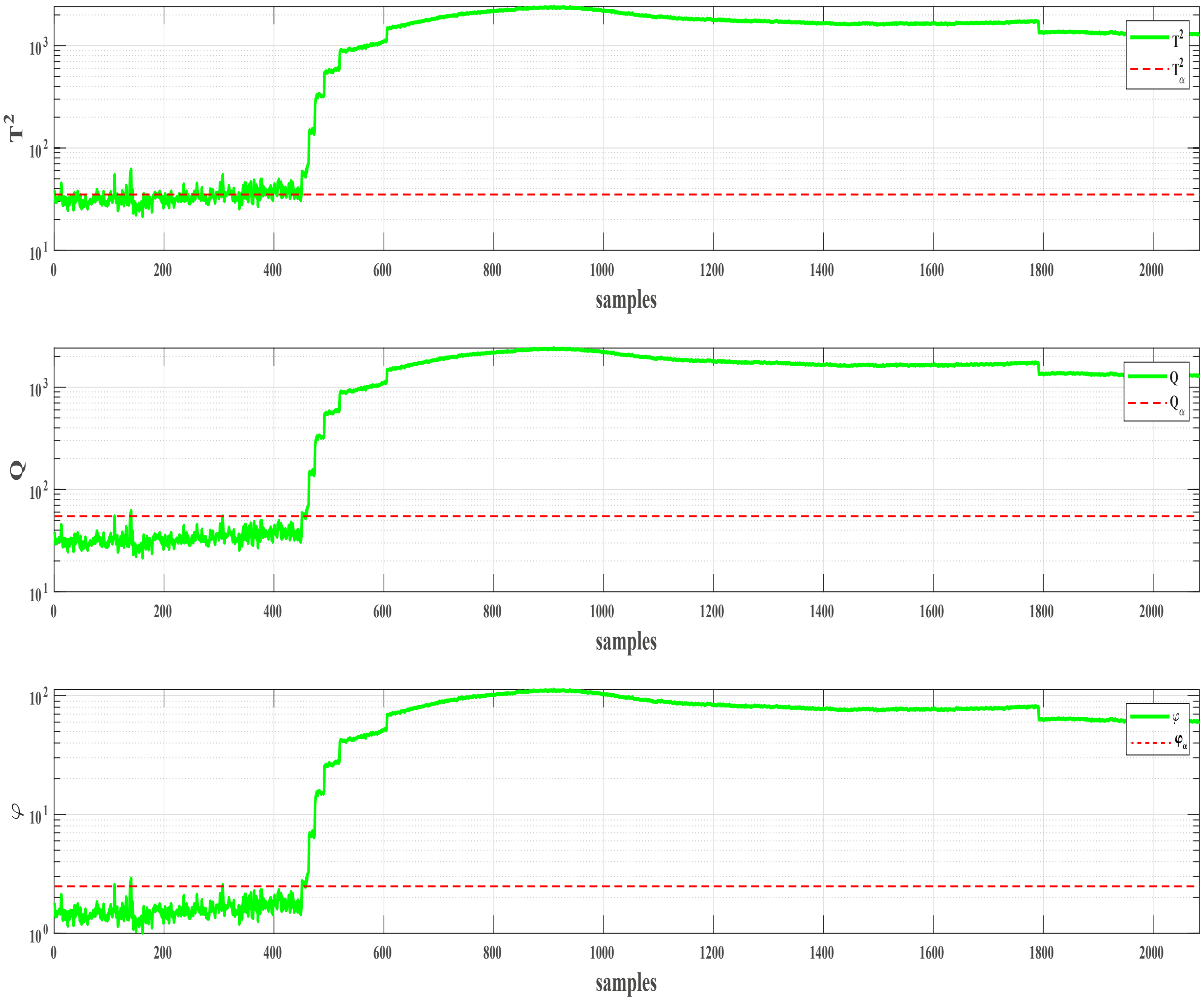


Fig. 5.10 RKPCA based correlation monitoring results of real process fault

space computations, it even leads the monitoring algorithm to converge to better performance. Table 5.8 reveals MDR results using PCA, KPCA, RKPCA to monitor the different faulty datasets. The monitoring performance is evaluated in terms of T^2, Q, and φ fault indices. The evaluation of MDR results can be classified into two cases:

5.2.3.3 Case 1: Abrupt faults

It is noticeable from Table 5.8 that the different FDD methods have successfully detected all the faults with different missed detection rates. Abrupt and intermittent faults (2,5,7) have been strongly detected with zero MDR using all three FDD techniques. Figures 5.11, 5.12, 5.15, and 5.17 display the monitoring results of the different abrupt faults contributed by fault indices T^2, Q, φ.

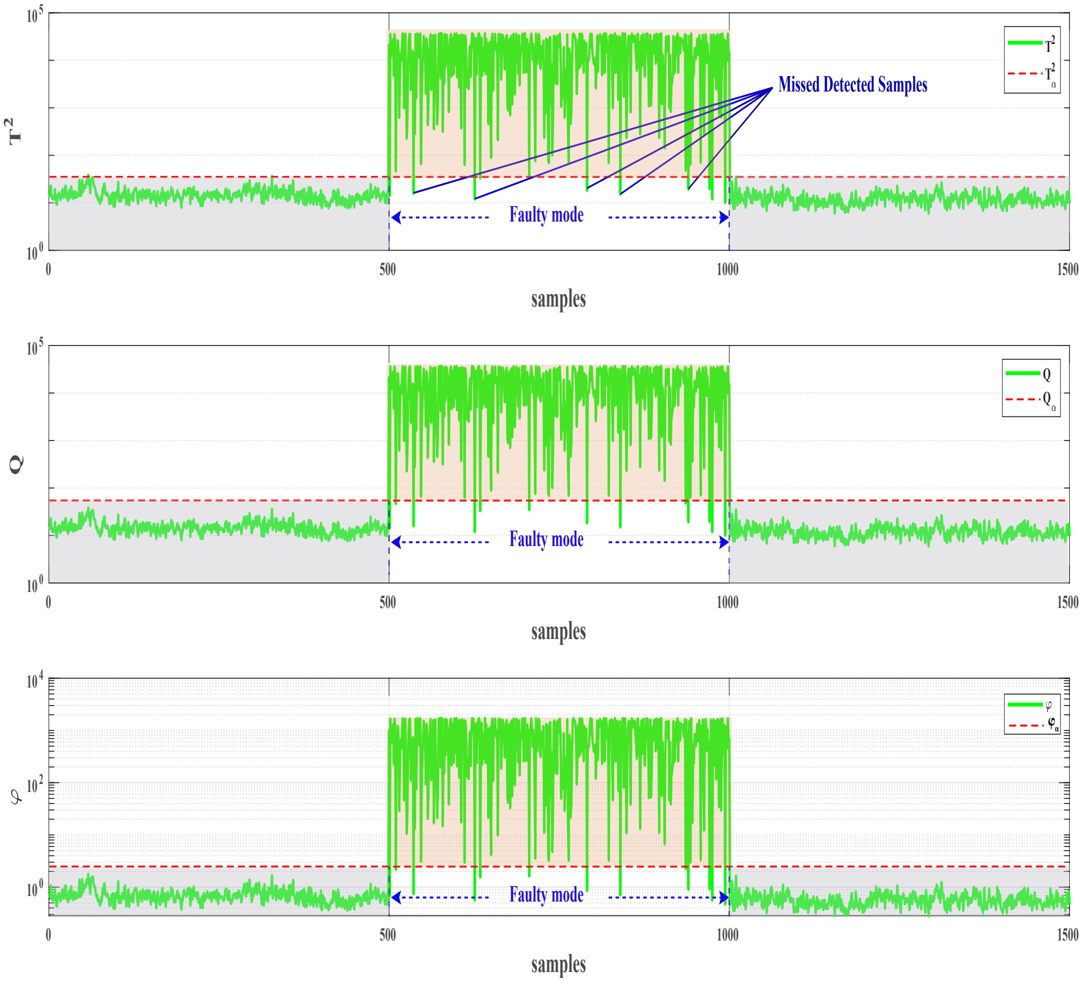


Fig. 5.11 RKPCA based correlation monitoring results of simulated fault1

5.2.3.4 case 2: Drift faults

In the real process fault, the occurrence time of the fault is expected to be at observation 450. So the evaluation of the fault starts at 450 s. From Table 5.8, it can be noticed that RKPCA-corr has detected the real process fault with less MDR compared to PCA and KPCA using all fault indices, while RKPCA has some difficulties in drift (incipient) simulated faults (fault 3 and 4). In fault 3, PCA made the best result meanwhile in fault 4 KPCA had less MDR, Although all the three have approximately devoted close results. The monitoring of drift faults is the most difficult fault due to small development in the fault magnitude over time, although our RKPCA was able to detect the faults with acceptable MDR. Figures 5.10, 5.13, 5.14, and 5.16 represent the monitoring plots of drift faults (real process fault and faults 3, 4, and 6 respectively) using different fault indices. The

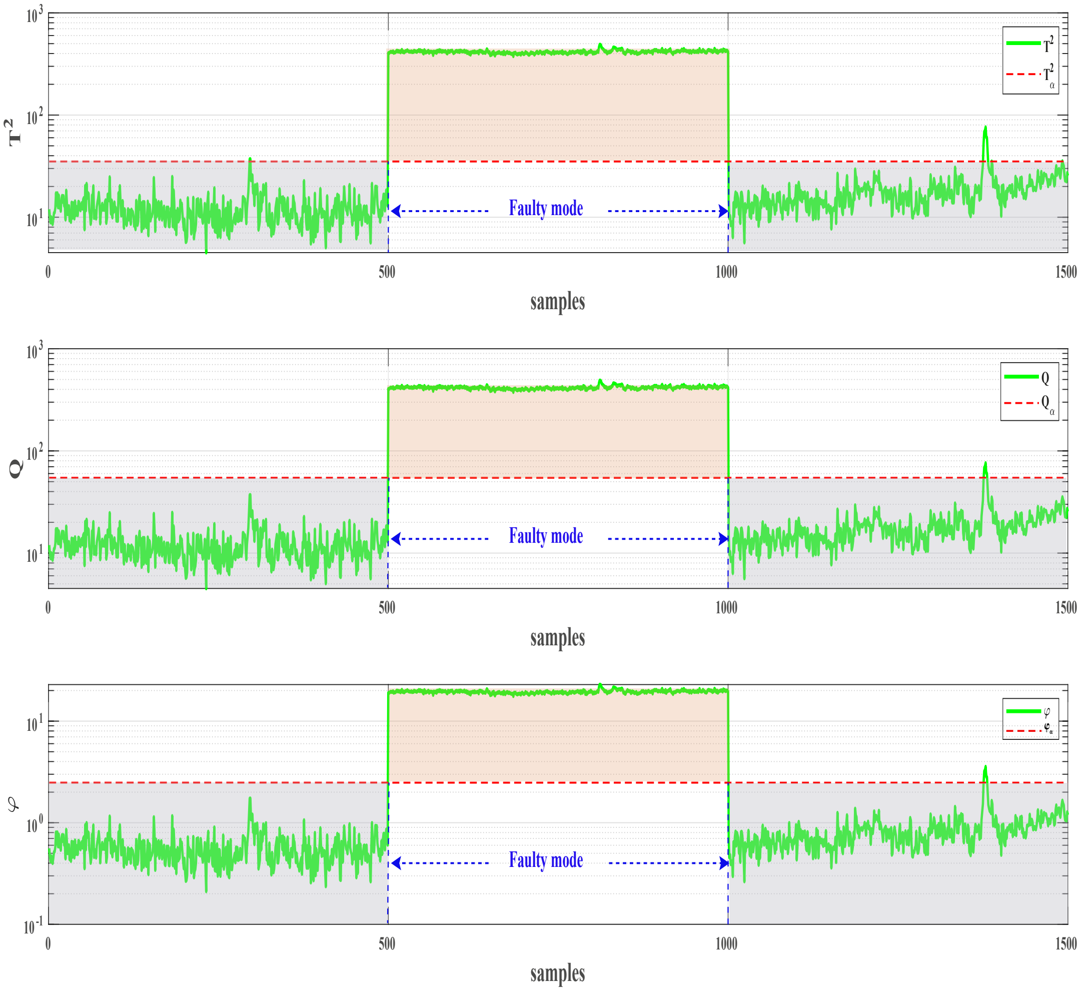


Fig. 5.12 RKPCA based correlation monitoring results of simulated fault2

last row of Table 5.8 displays the average MDR of each fault index in the PCA, KPCA, RKPCA. clearly, our RKPCA has the best results using T^2 and Q while KPCA has less MDR using φ. PCA has contributed the highest average FAR, and MDR compared to the nonlinear methods (KPCA, and RKCPA-corr) this is due to the fact that PCA assumes a linear relationship among the process variables. The linearity assumption will lead to degradation in the monitoring performance. Table 5.9 summarizes the results of the detection time delay of different faults using PCA, KPCA, RKPCA. It is clear that real fault and abrupt faults are easy to be detected just after one sample, every FDD scheme was able to detect it using the three fault indices. while the drift faults have a slight delay in the detection. This delay is due to slow development in fault about $\pm 4 \times 10^{-5}$ slopes. Although the faults have been detected before they reach 10%. The average results show that RKPCA contributed less DTD in Q index while PCA and KPCA have the same averages in T^2 and

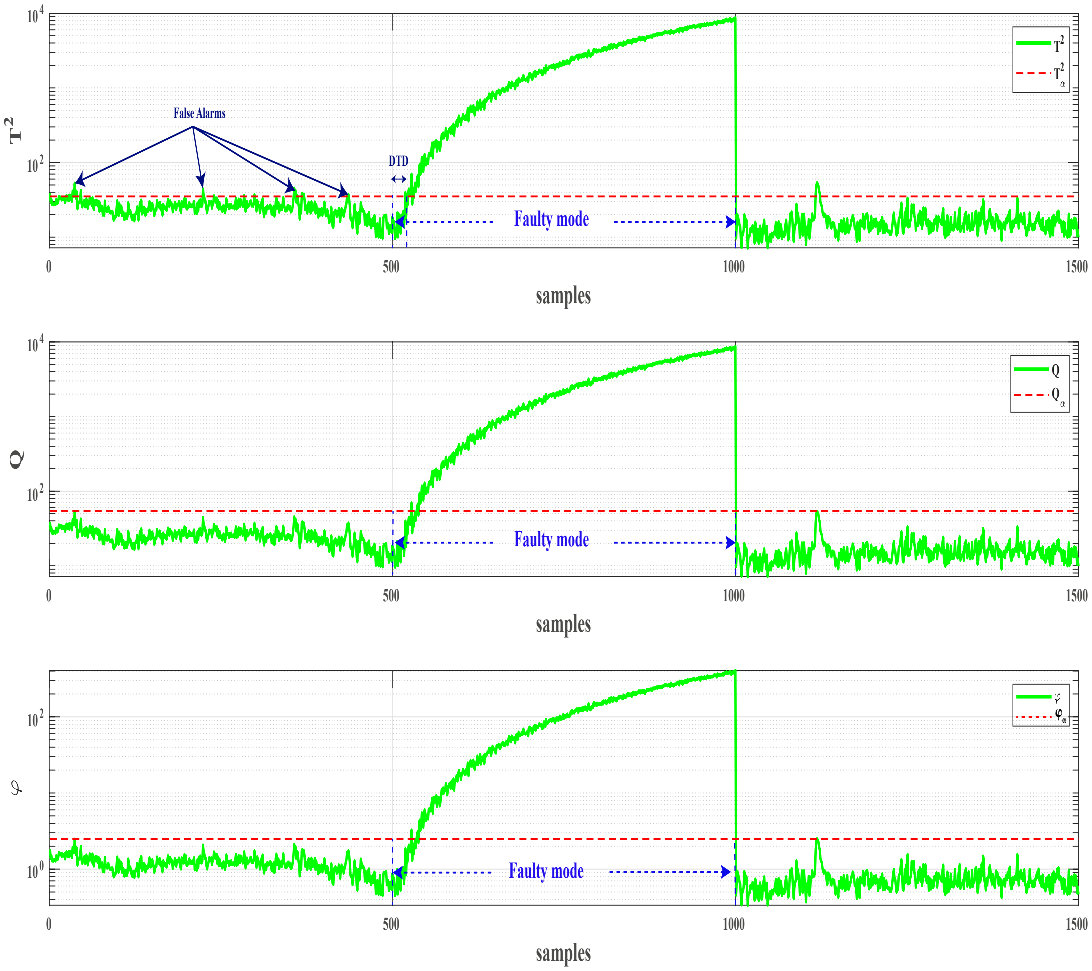


Fig. 5.13 RKPCA based correlation monitoring results of simulated fault3

φ.

5.2.4 RKPCA based cosine pairwise

To evaluate the efficiency of the proposed RKPCA based cosine pairwise in FDD and its ability to monitor nonlinear industrial processes with less computational time and storage space, it has been applied to monitor Ain Elkebira cement rotary kiln process. This RKPCA technique consists of three stages: the first is the reduction of the training data size. The data reduction is based on Independence of training observations. such that the independent observations are only kept and the dependent ones are eliminated due to the fact that they provide redundant information and have the same effect on the monitoring scheme. In the second step, The reduced dataset is used to build RKPCA scheme. The last step is to

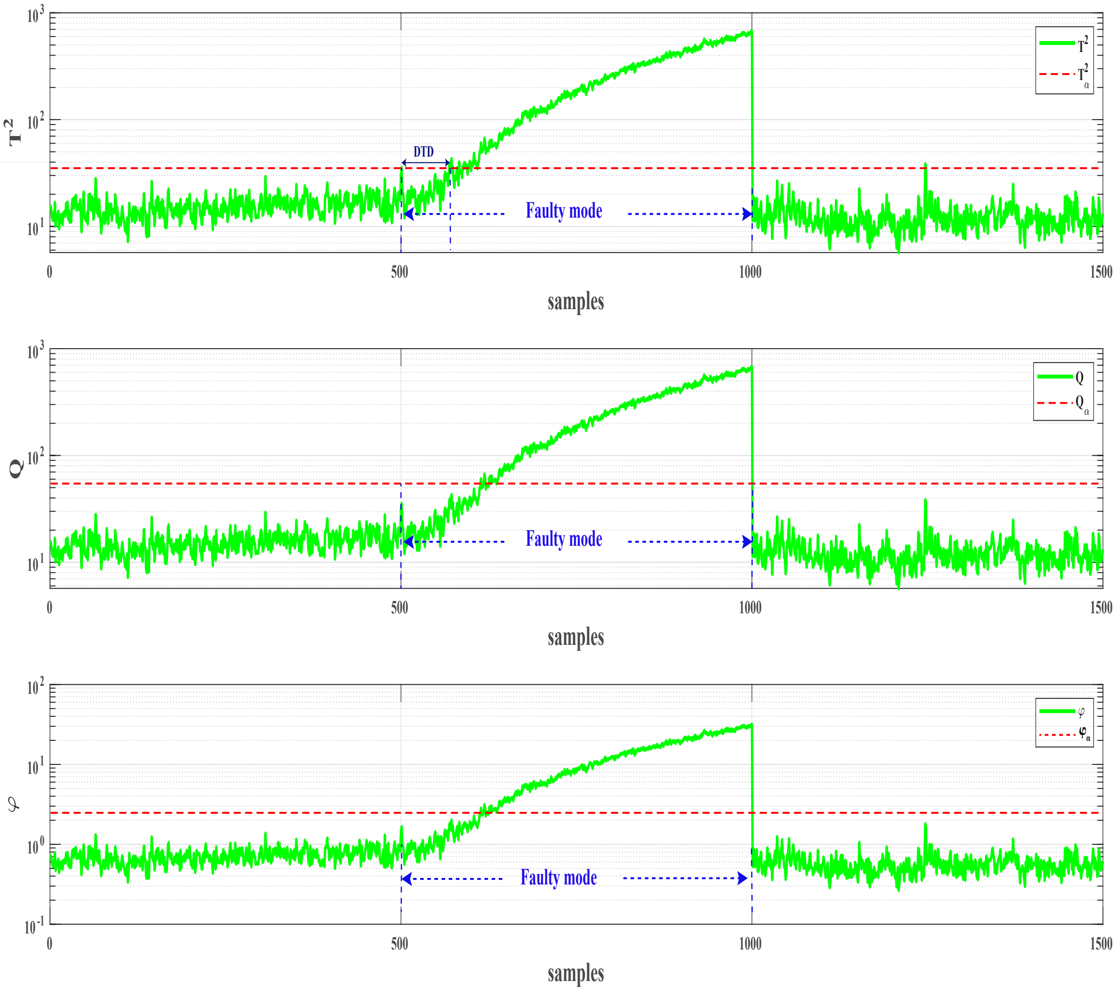


Fig. 5.14 RKPCA based correlation monitoring results of simulated fault4

monitor test the scheme based on FAR, MDR, and DTD.

5.2.4.1 Training dataset reduction

The determination of relatively independent observations is based on the cosine pairwise distance between samples, so that any observations with a cosine pairwise distance nearly equals 1 are considered as independent. Due to the noise among training dataset and error measurement it is recommended to select an interval around 1. To select an accurate independence interval, the cosine pairwise distance matrix has been computed. The cosine pairwise is in the range from 0.0306 to 1.763. The loss function $J(T^2, Q, \varphi, \epsilon)$ summarizes the average performance versus the independence interval ϵ. the values of ϵ start from zero and each time is increased to include another cosine distance. At each interval, a

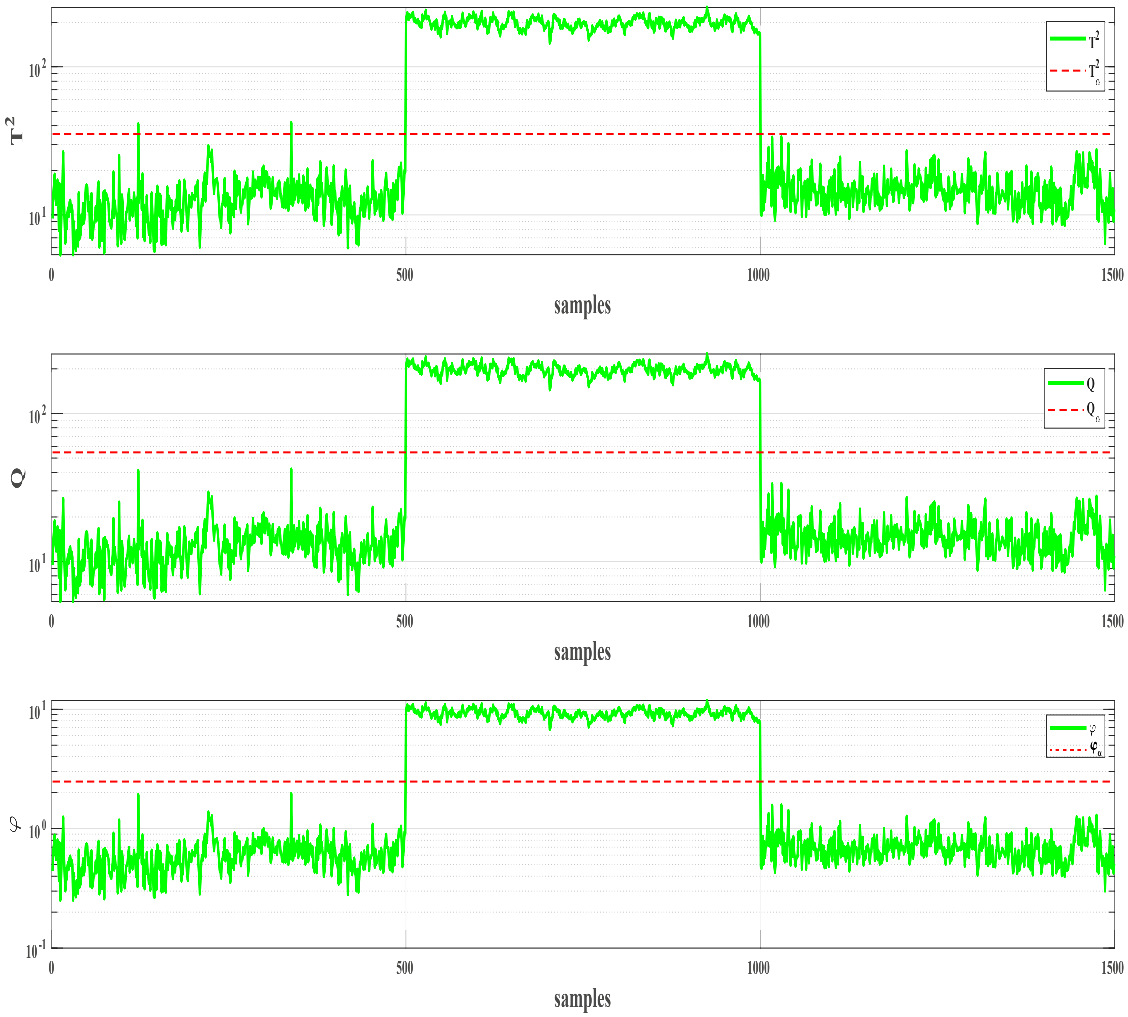


Fig. 5.15 RKPCA based correlation monitoring results of simulated fault5

reduced dataset is generated to be used as a training dataset for the KPCA algorithm. Performance is recorded to evaluate the value of $J(T^2, Q, \varphi, \epsilon)$. The value of ϵ that minimizes $J(T^2, Q, \varphi, \epsilon)$ is selected as optimal independence interval. Figure 5.18 displays the change of the loss function versus ϵ. From that figure the value $\epsilon = 4.205 \times 10^{-4}$ is the selected Independence interval. Table 5.10 shows the training observations number along with the KPCA execution time and the overall monitoring performance with different Independence intervals. It is clear that selecting $\epsilon = 1$ will take whole training observations thus no reduction and high computational time $T = 350s$. decreasing the interval surely will reduce the number of observations and the execution time while the monitoring slightly deteriorates then improved until it reaches its optimal performance at $\epsilon = 4.205 \times 10^{-4}$ after that when ϵ gets close to zero it reduces important observations. The remained samples will not be able to monitor the process.

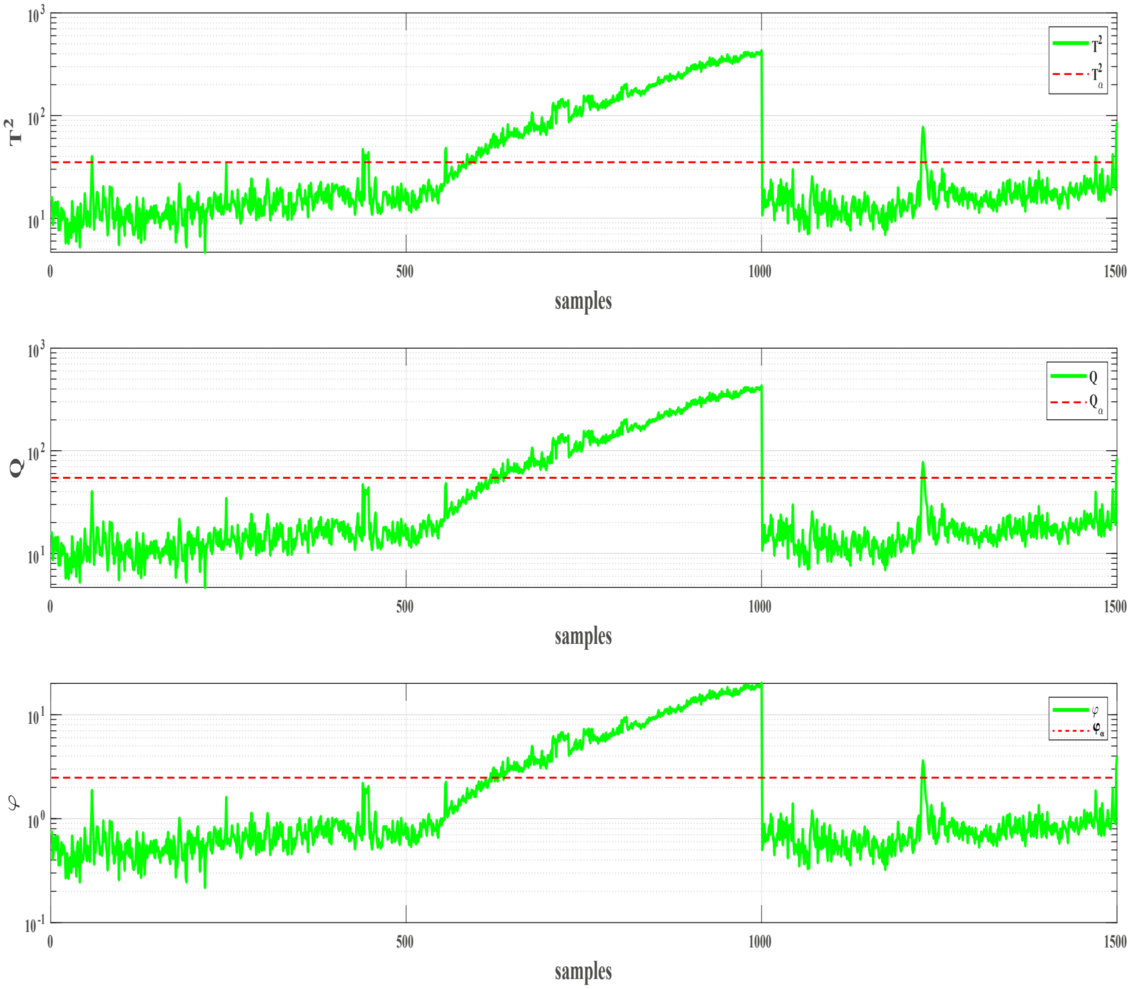


Fig. 5.16 RKPCA based correlation monitoring results of simulated fault6

Table 5.10 RKPCA performance versus ϵ

ϵ	Size of training data	RKPCA execution time (s)	J
1.00	768	350	8.1
1.00×10^{-1}	719	312	8.21
1.00×10^{-2}	682	277	8.27
1.00×10^{-3}	427	155	8.23
4.21×10^{-4}	232	80	**7.97**
2.30×10^{-5}	31	11	17.75

5.2.4.2 Results and discussion

Selecting $\epsilon = 4.205 \times 10^{-4}$ would reduce the training dataset from 768 to 232 observations or 69.79% reduction. The reduced training dataset $X_{red} \in \mathbb{R}^{232 \times 44}$ is used as training dataset to KPCA to monitor the process. RKPCA based cosine is compared to the ordinary KPCA, RKPCA based correlation, RKPCA based PCA [131] which reduced the

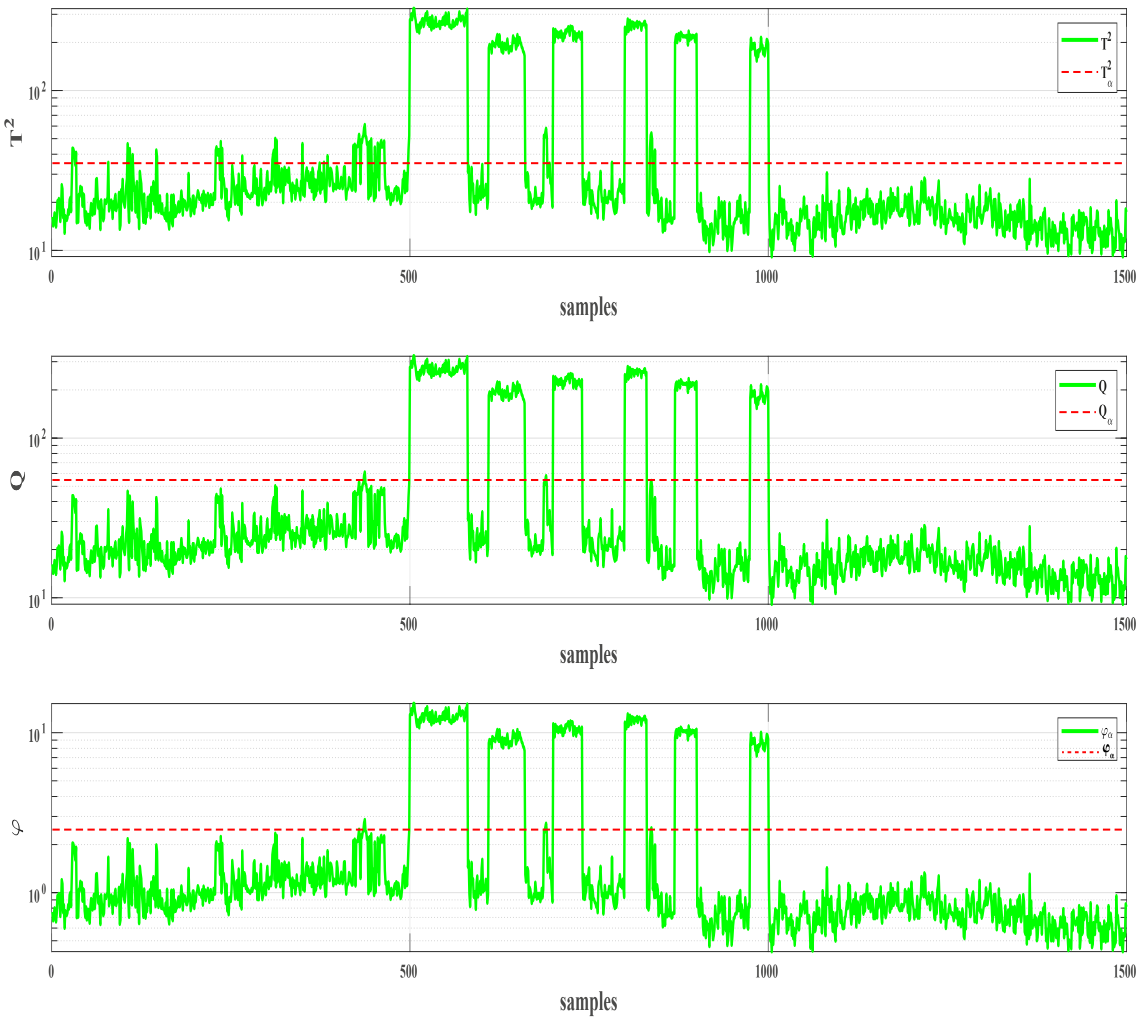


Fig. 5.17 RKPCA based correlation monitoring results of simulated fault7

original dataset to 31 observations, RKPCA based correlation RKPCA ED [10]. In both RKPCA-ED and RKPCA-corr, The training dataset is reduced to 131, and 513 respectively observation. A set of three simulated faults(abrupt, drift, and intermittent), and real process fault are used to test the proposed RKPCA techniques. RKPCA techniques aim to reduce the training dataset size while reserving the same statistical properties. So the statistical properties of the reduced datasets via the aforementioned RKPCA are evaluated. So it is required to investigate the means and standard deviations of the reduced training datasets to see whether they reserved the same statistical properties of the original dataset or there are changes. Figure 5.19 plots the means and the standard deviations of the 44 different variables of Ain El Kebira rotary kiln process of original and reduced datasets. The figure shows small changes in the means and standard deviations of the variables of the reduced dataset compared to the original one. It is clear small changes between the original and

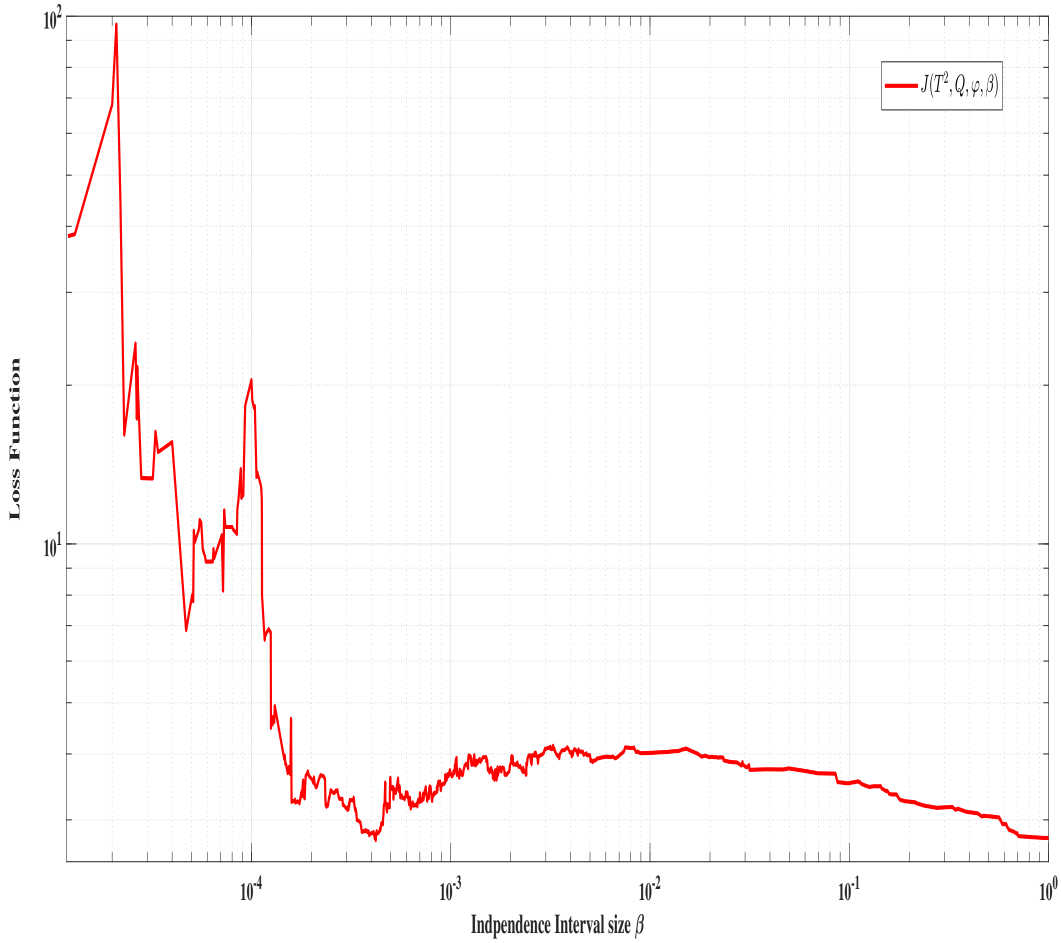


Fig. 5.18 Loss function $J(T^2, Q, \varphi, \epsilon)$ plot of RKPCA-cos

reduced datasets won't lead to changes in the statistical characteristics of the process ,thus, the reduced dataset can monitor the process instead of the whole dataset. RKPCA based PCA retains a normalized reduced training dataset ,thus, it cannot be compared to other techniques in terms of means and standard deviations. So the means and standard deviations are not enough to see the changes in the statistical properties. Kullback Leibler divergences (KLD) of the reduced datasets are computed to compare the reduced training datasets. Figure 5.20 presents the KLD values of the aforementioned RKPCA methods. All the reduced dataset shows less than 1 KLD values in the different 44 variables this can be considered that all the RKPCA methods keep the same statistical properties of the original training dataset. It can be not seen that the RKPCA based PCA [131] have the largest KLD values while the proposed RKPCA (RKPCA-ED, RKPCA-corr, and RKPCA-cos) have KLD varies from 0.1 to 0.5.

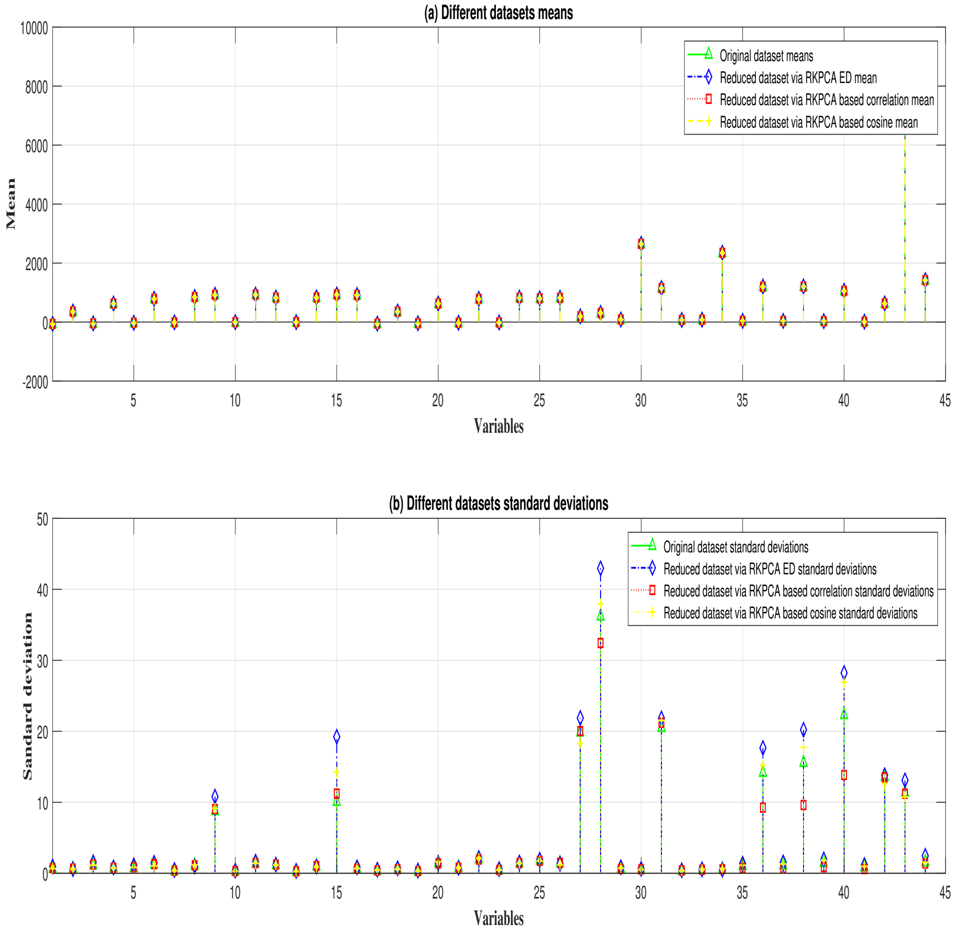


Fig. 5.19 Means and standard deviations of cement rotary kiln different datasets

The reduced data $X_{red} \in \mathbb{R}^{232\times44}$ has been used to train KPCA model and yield the principal and residual components in the feature space in order to compute the fault detection indices and their corresponding thresholds. $CPV(l) = 80\%$ has been used to determine the number of retained principal components. To validate the RKPCA model, a test dataset of 11000 has been used to evaluate FAR contributed by the three fault indices. FAR rate should be minimal (less 5% in order to accept the model). The monitoring of these faulty datasets is evaluated in terms of FAR, MDR and DTD and J multi-objective function which summarizes the monitoring performance (FAR, MDR and DTD contributed by T^2, Q, and φ in one index), and then compared to the results obtained by applying the KPCA and the aforementioned RKPCA techniques. Table 5.11 reveals FAR, MDR, DTD, contributed by the different fault indices T^2, Q, and φ via the ordinary KPCA, RKPCA based correlation, RKPCA based PCA [131], RKPCA based ED [10], and RKPCA

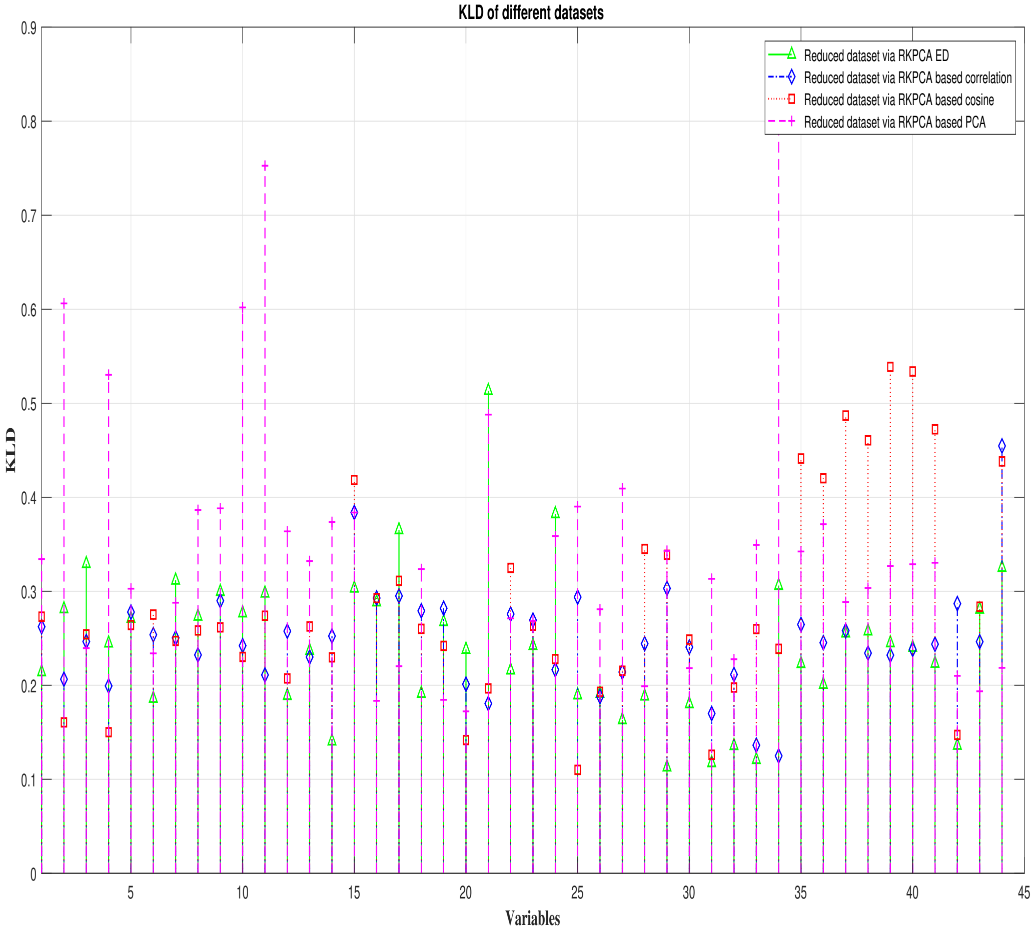


Fig. 5.20 KL divergence values of cement plant reduced datasets using different RKPCA approaches

based cosine methods of the different faulty datasets. The last row indicates the average values of multi-objective index $J(.)$ that summarizes FAR, MDR, DTD of each fault index of the different fault indices. J is used to determine the best results. Thus the smallest values of J are considered as best results and any FAR, MDR, and DTD contributed by T^2, Q, φ are considered best J performances and writing in bold. Starting with the real process fault, using T^2 index RKPCA-ED has 0.00% FAR while other RKPCA techniques have FAR around a small amount of false alarms (except RKPCA-corr which has 33.70), all the techniques have detected the fault with almost 0% MDR and 1s DTD. Q index RKPCA-PCA provided best FAR, RKPCA-cos with best MDR and DTD. with φ RKPCA-corr has 0.40% FAR, RKPCA contributes 0.06% MDR and detecting the fault after 1s. Fig.5.21 plots T^2, Q, and φ of the real process fault using RKPCA based cosine. In the abrupt fault,

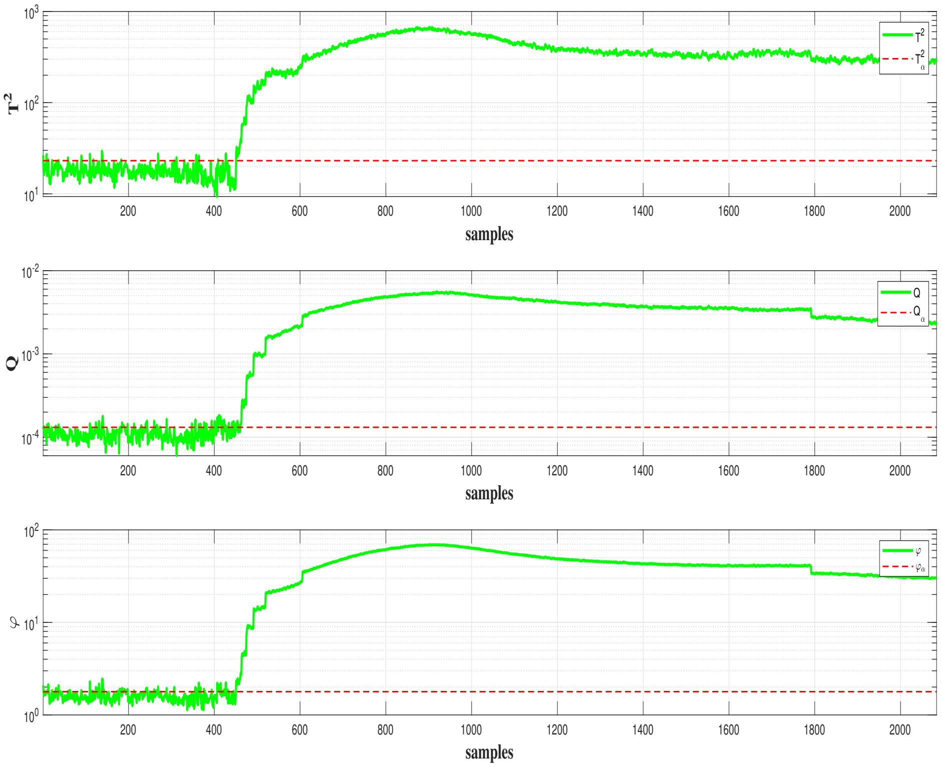


Fig. 5.21 RKPCA based cosine monitoring results of real process fault

RKPCA detected the fault using three indices with 0% FAR, 0% MDR, and 1s DTD. The other techniques have a small amount of FAR and MDR but the detection is after 1s. The drift fault is detected with RKPCA-cos contributing 0.20% using T^2 and 11.40% MDR, Q index detected the fault after 6s and with RKPCA PCA using Q index 0.20% FAR(similar to the ordinary KPCA). The intermittent fault is detected with 0% MDR and 1s DTD using all techniques with small amounts of FAR. The last row shows the multi-objective index RKPCA based PCA contributed best results using T^2 index meanwhile RKPCA based cosine gives the best performance using Q and φ. The index J summarizes the FAR, MDR, DTD in one indicator. It is clear in T^2 that all the proposed RKPCA have outdone the ordinary KPCA, Using Q and φ indices, RKPCA based cosine has contributed the best result. To reduce the time complexity to $O(N^2)$ it is required to reduce the dataset to less than 84. the proposed techniques are not able to reduce the training data to the required size thus the time complexity is $O(r^3)$. The proposed techniques have provided great performance in terms of fault detection minimizing FAR, MDR, DTD percentages using the three fault indices (overcome KPCA) in one hand. On the other hand, the proposed methods were able to solve the high computation issue of the ordinary KPCA proving the lowest computation time and space.

Table 5.11 The monitoring performance of the different fault indices and via different techniques in the cement plant process

Faults	RKPCA based correlation			RKPCA based PCA [131]			RKPCA based ED [10]			RKPCA based cosine			KPCA		
	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ
Real process fault	32.70|0.00|1	10.46|0.55|1	0.40|0.30|1	4.89|0.06|2	0.67|0.68|11	65.03|0.00|1	0.22|0.79|1	5.12|0.48|1	0.00|2.19|1	4.67|0.06|1	7.20|0.36|1	6.30|0.06|1	8.01|0.60|1	6.68|0.60|1	2.00|0.40|1
SimFault1	0.20|1.99|1	0.80|1.59|1	0.00|2.19|1	5.60|6.00|1	0.00|0.00|1	10.80|0.00|1	0.00|3.79|1	3.5|1.19|1	0.00|0.00|1	0.00|0.00|1	0.40|0.00|1	0.00|0.00|1	1.80|7.60|1	0.80|0.00|1	0.60|0.00|1
SimFault2	1.90|17.36|8	0.90|16.65|25	0.60|20.55|25	5.80|11.60|7	0.20|5.20|25	11.00|3.20|7	0.00|6.80|29	1.90|5.90|7	0.00|7.40|29	0.20|11.40|41	0.60|4.20|6	0.20|5.60|16	1.80|12.20|47	0.20|4.40|18	0.60|5.80|28
SimFault3	5.80|0.00|1	0.88|0.00|1	0.16|0.00|1	24.00|0.00|1	2.67|0.00|1	52.00|0.00|1	0.00|0.00|1	3.79|0.00|1	0.00|0.00|1	2.00|0.00|1	1.67|0.00|1	1.67|0.00|1	7.33|0.00|1	4.67|0.00|1	4.67|0.00|1
J	4.01	4.72	4.34	**3.00**	4.49	6.00	4.07	2.01	4.00	4.46	**1.35**	**2.16**	7.22	2.93	3.86

5.3 Tennessee Eastman process

5.3.1 Process description

The TE process was first proposed in [141] by Downs and Vogel (1993) describing an industrial chemical process to develop and evaluate plant-wide control strategy and multivariate process monitoring techniques. This benchmark has been widely utilized by many researchers to test and validate their fault detection and diagnosis algorithms. The system consist of five principle unit operations: Reactor, Condenser, recycle Compressor, a vapor-liquid Separator, product Stripper. Figure 5.22 illustrates the different units of TEP. The process consists of 52 variables [142] with 12 manipulated variables$XMV(1)$ to $XMV(12)$ and 41 measured variable $XMEAS(1)$ through $XMEAS(41)$. Table 5.12 lists the 12 manipulated variables and Table 5.13 presents the 41 measured variables. The process data has training dataset and and 21 programmed faults (shown in Table 5.14) with 960 samples, the faults have been introduces after 160 samples.

Table 5.12 TE process manipulated variables

XMV	Description
1	Valve position feed component D (stream 2)
2	Valve position feed component E (stream 3)
3	Valve position feed component A (stream 1)
4	Valve position feed component Aand C stream 4
5	Valve position compressor re-cycle
6	Valve position purge (stream 9)
7	Valve position underflow separator stream10
8	Valve position underflow stripper (stream 11)
9	Valve position stripper steam
10	Valve position cooling water outlet of reactor
11	Valve position cooling water outlet of separator
12	Rotation of agitator of reactor

5.3.2 RKPCA based ED data reduction

The proposed RKPCA-ED approach has been applied to TEP benchmark. The training data set under the healthy operating condition is reduced by eliminating the similarity observations affecting the monitoring performance. First, all the Euclidean distances between the n observations are calculated and sorted in descending order to have $\frac{n(n-1)}{2}$ distances. Similar observations normally have Euclidean distance equal to zero but due to the noise and the uncertainties, we could take distances greater than zero. All observations that have

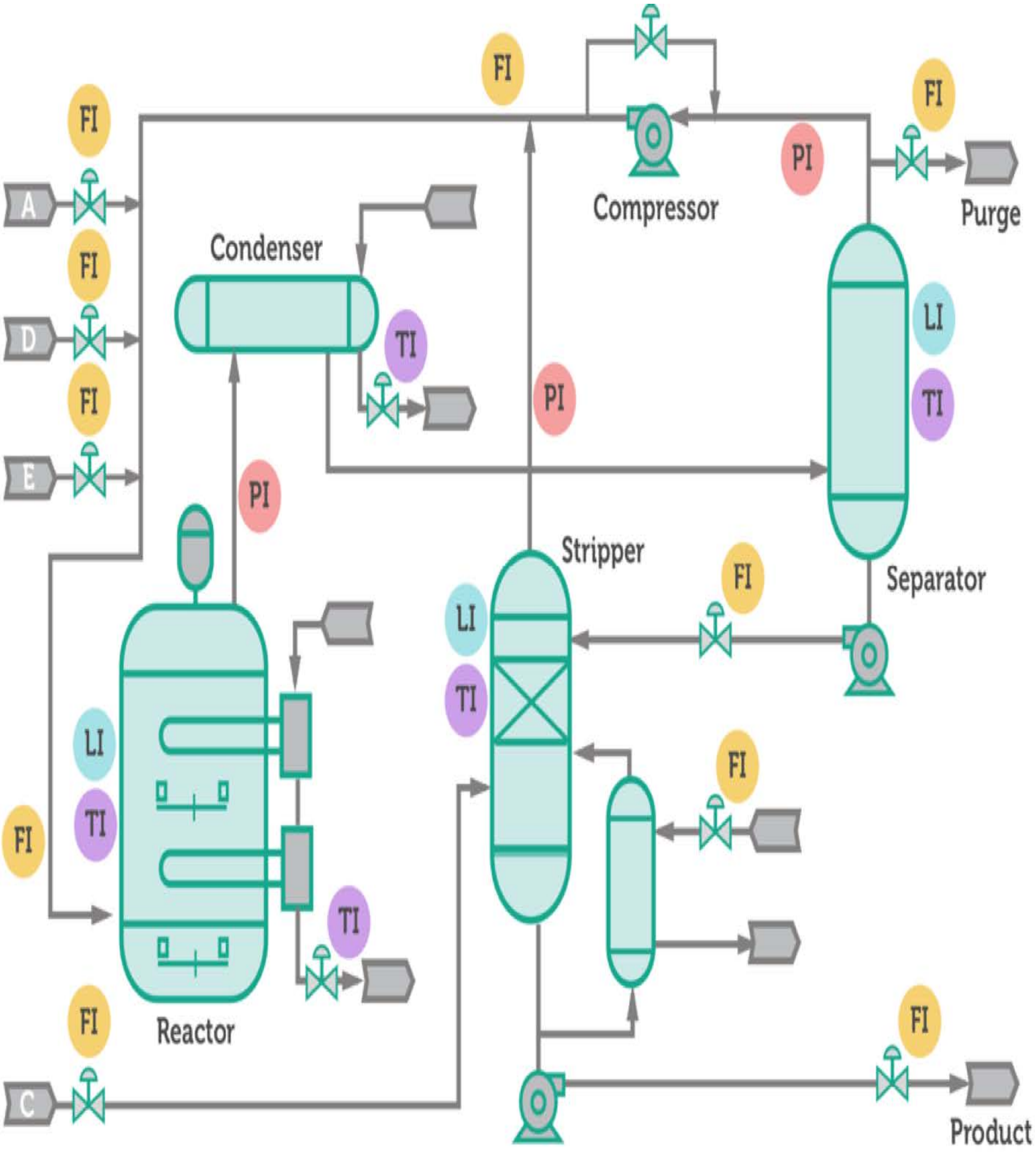


Fig. 5.22 Diagram of Tennessee Eastman process

Euclidean distances less or equal similarity threshold can be considered as similar observations so they will be represented by only one observation. After neglecting the similar observations, a reduced training data set is generated. After computing the distance matrix and sort its elements in increasing order it is found that the distances are varying from 3.012 to 16.6011. To determine the best similarity threshold only distances varying from 3.012 to 8.7 are taken in concentration. The optimal similarity threshold for monitoring TEP process is determined by optimizing the loss function. three loss functions $J(T^2)$, $J(Q)$, $J(\varphi)$ are computed, then $J(\beta, T^2, Q, \varphi)$ is obtained. The optimal value of J_{T^2} and

Table 5.13 TE process measured variables description

XMEAS	Description
1	Feed flow component A (stream 1)
2	Feed flow component D (stream 2)
3	Feed flow component E (stream 3)
4	Feed flow components A and C (stream 4)
5	Recycle flow to reactor from separator (stream 8)
6	Reactor feed (stream 6)
7	Reactor pressure
8	Reactor level
9	Reactor temperature
10	Purge flow (stream 9)
11	Valve position cooling water outlet of separator
12	Separator level
13	Separator pressure
14	Separator underflow (liquid phase)
15	Stripper level
16	Stripper pressure
17	Stripper underflow (stream 11)
18	Stripper temperature
19	Stripper steam flow
20	Compressor work
21	Reactor cooling water outlet temperature
22	Condenser cooling water outlet temperature
23 to27	Concentration in Reactor feed (stream 6),
28	Components A through F
29 to35	Concentration in Purge (stream 9),
36	Components A through H
37 to	Concentration in stripper underflow (stream 11),
41	Components D through H

J_Q corresponds to $\beta = 6.5$ while for J_ϕ is $\beta = 6.9$. Thus $\beta = 6.5$ corresponds to the overall loss function optimal value. $\beta = 6.5$ has been chosen to produce the training data set with size $l = 270$ instead of the original size $n = 960$, this reduced data set is used to build up a RKPCA model to monitor TEP process and to compute the fault indices thresholds T^2_α, Q_α, and φ_α.

5.3.3 RKPCA based correlation training dataset reduction

The selection of correlation threshold to eliminate the correlated observations and build a reduced dataset is based on one of the sorted correlation coefficients. The first correlation coefficient is selected as correlation threshold, the training dataset is reduced based on that threshold and the monitoring performance is recorded (FAR, MDR, and DTD

Table 5.14 Tennessee Eastman Process fault description

No	Fault description	Type
IDV(1)	A/C-ratio of stream 4, B composition constant	Step
IDV(2)	B composition of stream 4, A/C-ratio constant	Step
IDV(3)	D feed (stream 2) temperature	Step
IDV(4)	Cooling water inlet temperature of reactor	Step
IDV(5)	Cooling water inlet temperature of separator	Step
IDV(6)	A feed loss (stream 1)	Step
IDV(7)	C header pressure loss (stream 4)	Step
IDV(8)	A/B/C composition of stream 4	Random
IDV(9)	D feed (stream 2) temperature	Random
IDV(10)	C feed (stream 4) temperature	Random
IDV(11)	Cooling water inlet temperature of reactor	Random
IDV(12)	Cooling water inlet temperature of separator	Random
IDV(13)	Reaction kinetics	Drift
IDV(14)	Cooling water outlet valve of reactor	Sticking
IDV(15)	Cooling water outlet valve of separator	Sticking
IDV(16)	(unknown) deviations of heat transfer within stripper (heat exchanger)	Random
IDV(17)	(unknown) , deviations of heat transfer within reactor	Random
IDV(18)	(unknown), deviations of heat transfer within condenser	Random
IDV(19)	(unknown), recycle valve of compressor underflow separator (stream 10), underflow stripper (stream 11) and steam valve stripper	Sticking
IDV(20)	(unknown)	Random
IDV(21)	The valve for Stream 4 was fixed at the steady state position	Constant

contributed by the fault indices). Each time, the correlation threshold is selected as one of the correlation coefficients to reduce the training observations and record the performance. The performance of the RKPCA is evaluated using loss functions of three fault indices and their average loss function presented in section 4.6.2. Figure 5.23 shows loss function plot the optimal point is at $\gamma = 0.549$ which is able to reduce the dataset to 216 observations(22.5$\%$).

5.3.4 RKPCA based cosine training dataset reduction

The cosine pairwise distances between observations are computed in order to determine the dependent samples that have pairwise cosine distance near to one. Loss function $J(T^2, Q, \varphi, \epsilon)$ is evaluated to determine the accurate range size of ϵ. The performance is recorded in terms of FAR, MDR, and DTD with desired FAR and MDR values equal to 5% and 10 samples desired DTD. The optimal value of $J(T^2, Q, \varphi, \epsilon)$ is 9.47 at $\epsilon = 2.34 \times 10^{-3}$. The selected value of ϵ reduces the training dataset to 825 observations, which means more that 14% reduction. Figure 5.24 plots loss function $J(T^2, Q, \varphi, \epsilon)$ versus ϵ and indicates the optimal value of ϵ that minimizes loss function.

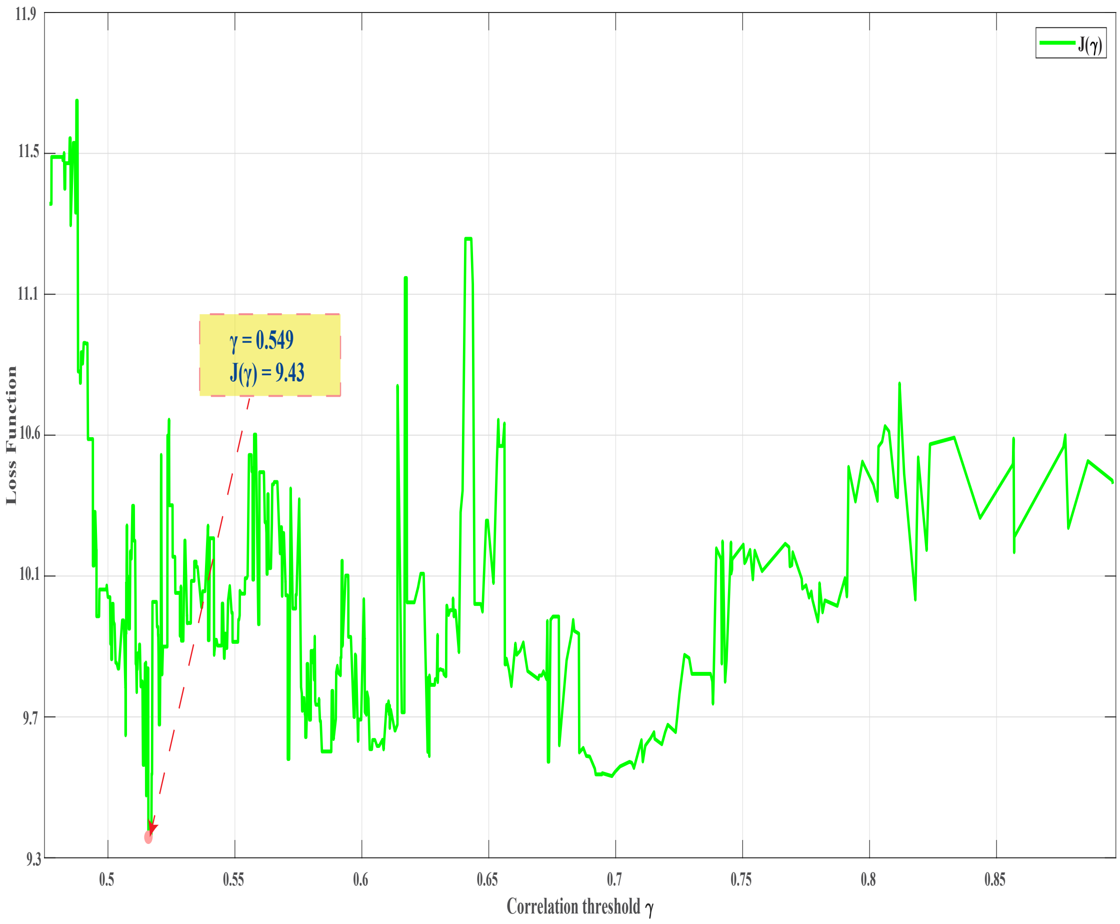


Fig. 5.23 RKPCA based correlation TEP loss function

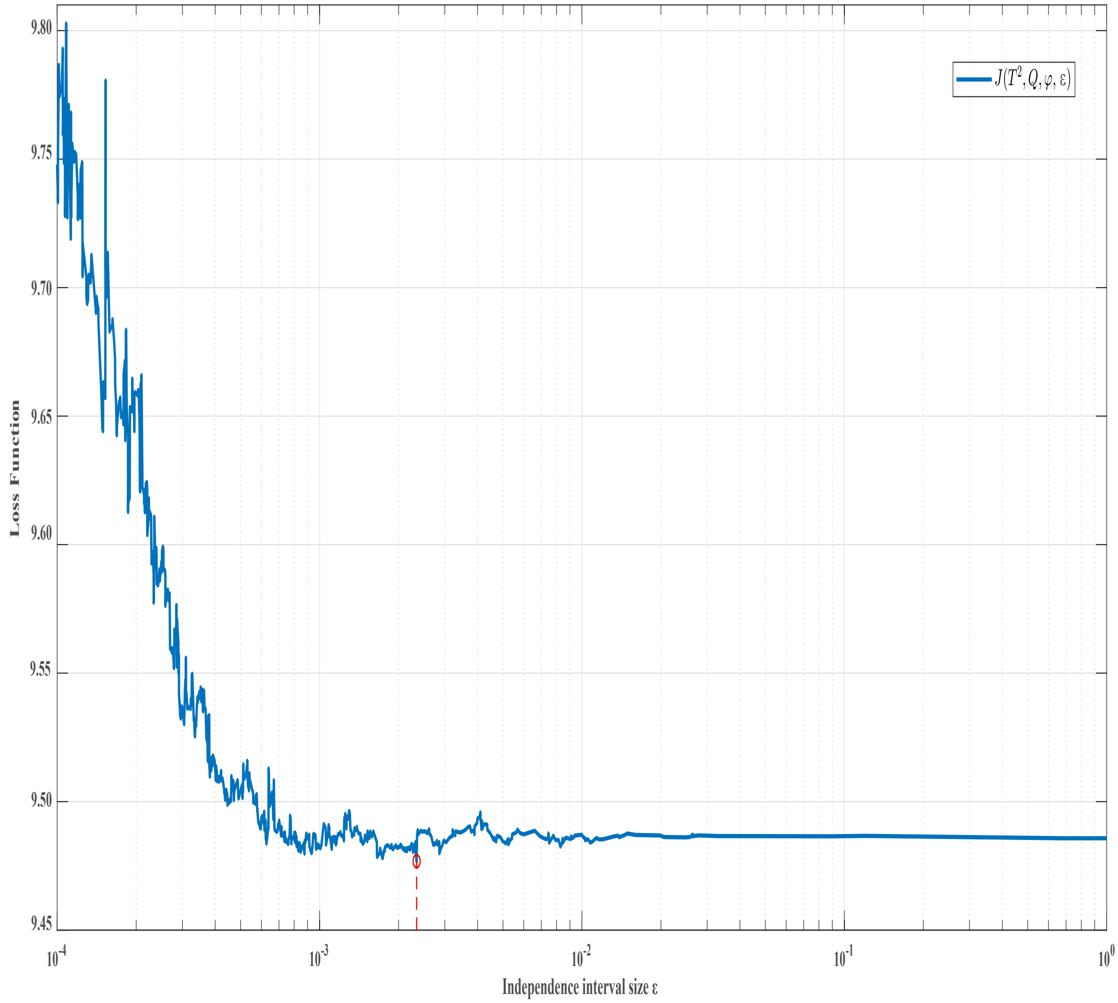


Fig. 5.24 TEP dataset loss function $J(T^2, Q, \varphi, \epsilon)$ plot using RKPCA based cosine

5.3.5 Results and discussion

The proposed methods (RKPCA based ED, RKPCA based correlation, and RKPCA based cosine) are applied to monitor TEP guarantee their ability to monitor more complex industrial processes. The results of the monitoring are recorded in terms of FAR, MDR, DTD, and multi-objective function J that summarizes FAR, MDR, DTD rates of each index (T^2, Q, and φ). These results are compared to the conventional KPCA method, RKPCA based PCA technique proposed by Harkat et al. [131].

Before evaluating the monitoring results, the reduced dataset via the aforementioned RKPCA methods are compared to the original dataset in terms of means, standard deviation, and Kullback Leibler divergence of each variable to ensure that the reduced datasets preserve the same statistical characteristics of the original dataset. Figures 5.25 plot the

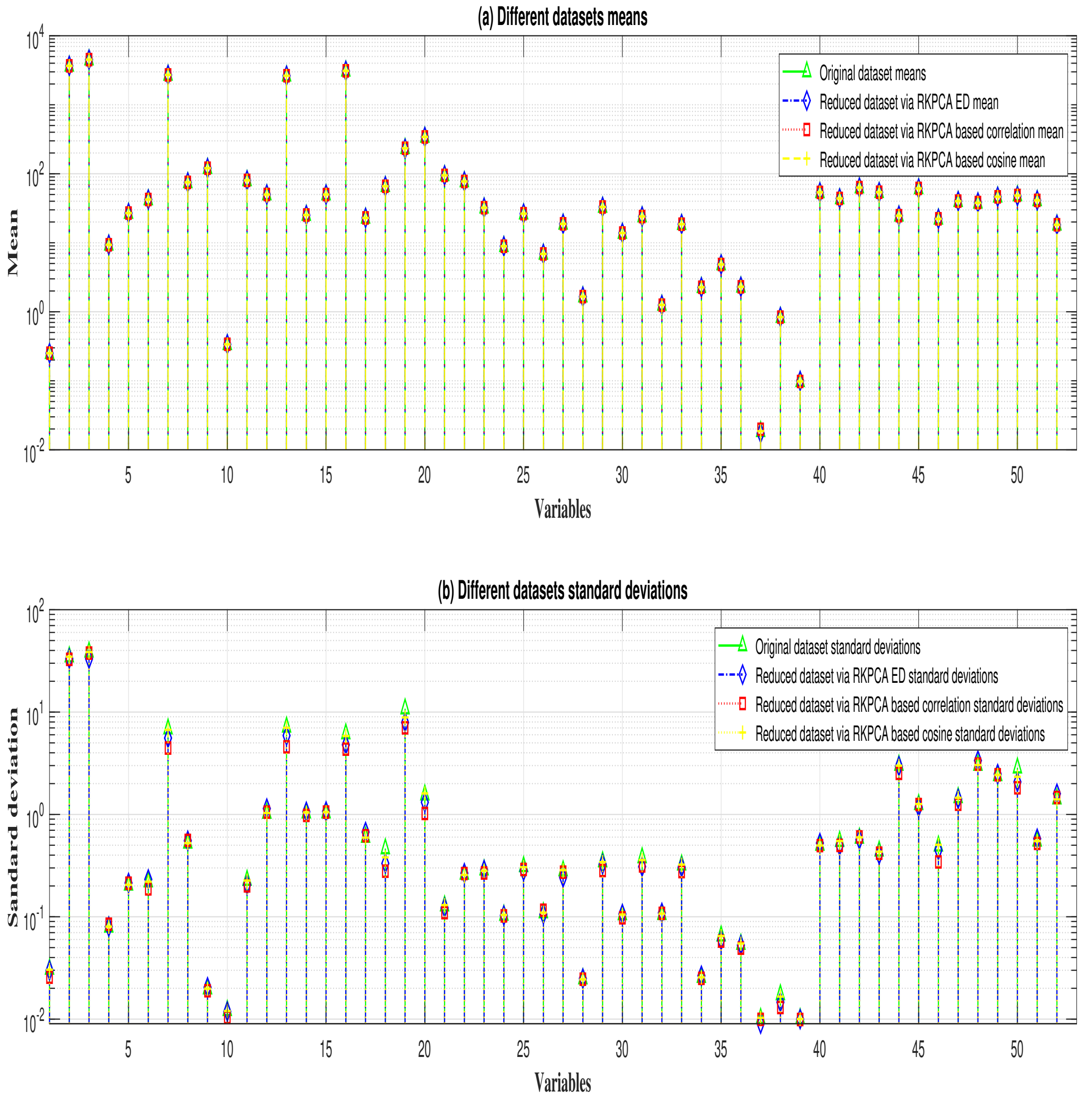


Fig. 5.25 Means and standard deviations of TEP different datasets

means and the standard deviations of 52 variables of original and reduced TEP datasets (except RKPCA based PCA due to the fact that this technique retains normalized dataset). The reduced number of observations leads to small changes in the statistical characteristics of the process. It is noted there are small changes in the means and variances of the reduced datasets but the small changes are not clear so KLD of reduced datasets is computed and plotted in figure 5.26. KLD of the proposed techniques is computed using the original non-normalized training dataset. In RKPCA based PCA KLD is calculated using a normalized training dataset.

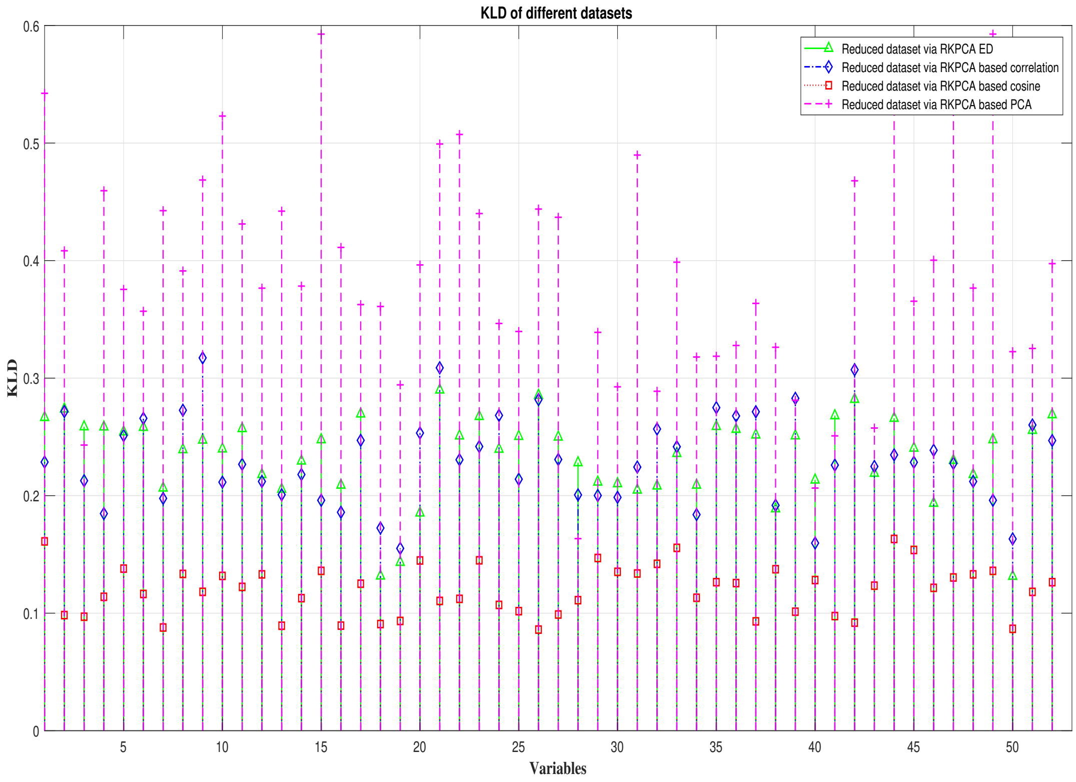


Fig. 5.26 KL divergence values of TEP reduced datasets using different RKPCA approaches

It can be seen that RKPCA based cosine has the smallest KLD rates while RKPCA based PCA have large KLD, the other RKPCA methods have an acceptable rate which means that they all reduced RKPCA are almost kept the same statistical properties of the original training dataset (slight change in RKPCA based PCA). The reduced training datasets are used to build a reduced kernel matrix in order to determine the principal components at $CPV(l) = 85\%$. The performance of 21 faults dataset is compared to the conventional KPCA in terms of FAR and MDR contributed by T^2, Q, and φ then $J = \frac{FAR(.)}{5} + \frac{MDR(.)}{5} + \frac{DTD}{10}$ is used to determine the best performance in each fault. The best J results are highlighted in bold to present the best results. Table 5.15 presents the FAR, MDR, and DTD results of 21 TEP faults contributed by T^2, Q, and φ of KPCA, RKPCA based PCA, RKPCA based ED [10], RKPCA based correlation, and RKPCA based cosine techniques. the last row is the averages of $J(.)$ of each index. RKPCA based correlation has best $J(T^2) = 3.11$ means that it has contributed best performance using T^2 index. RKPCA based cosine contributed $J(Q) = 2.83$ and $J(\varphi) = 2.63$ so it has the best results using Q, and φ indices. Coming to the overall performance J, RKPCA

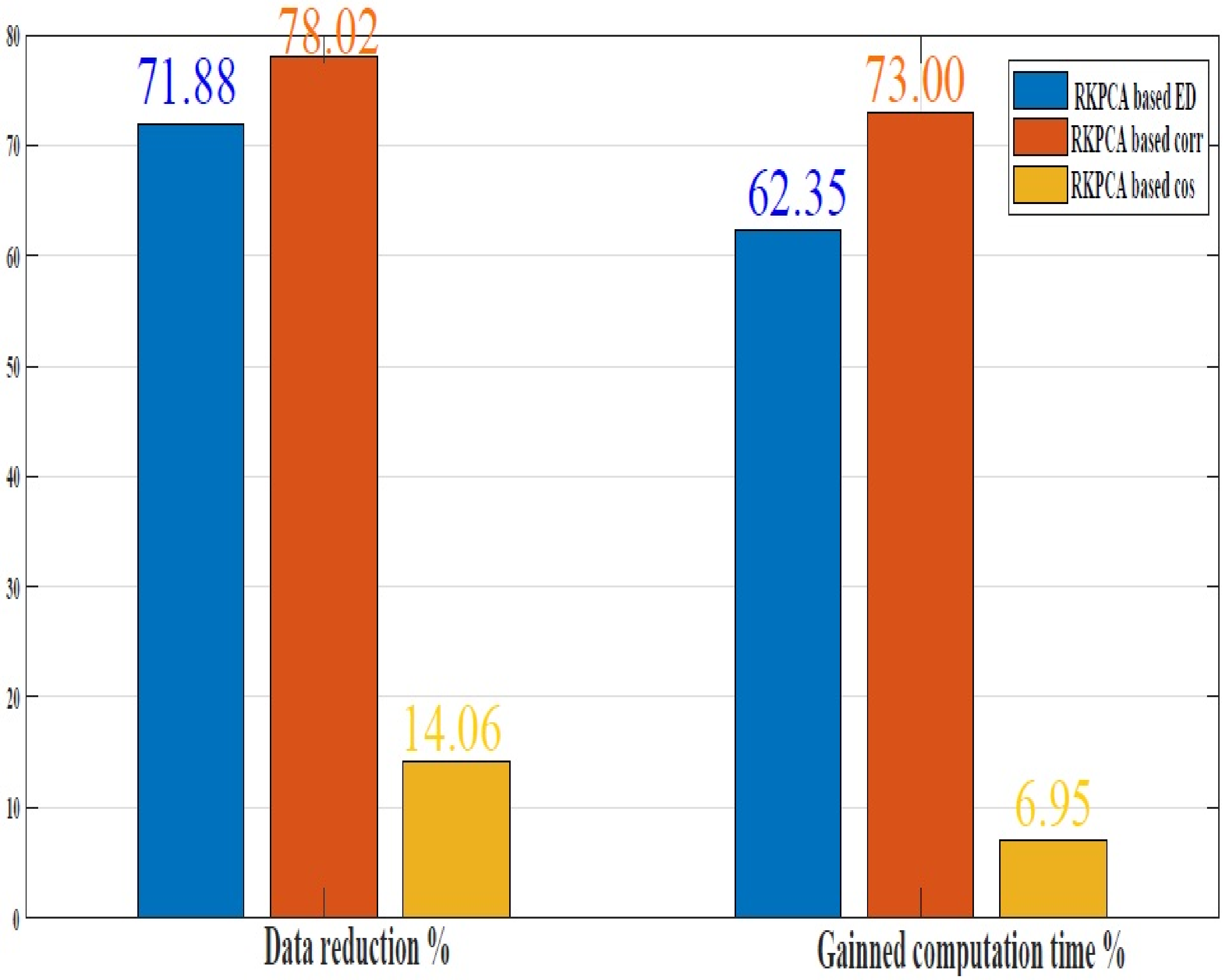


Fig. 5.27 TEP data reduction and gained computation time percentages

based correlation has outdone all the ordinary KPCA and the other RKPCA techniques with $J = 9.43$. The ordinary KPCA has $J = 9.87$, RKPCA based PCA $J = 15.74$, and RKPCA ED comes with $J = 10.10$. Clearly that RKPCA based PCA comes with the worst performance, as compression between RKPCA based correlation and RKPCA based PCA, both techniques remove the correlated observations in order to reduce the training dataset size but RKPCA based PCA eliminates all the correlated observations (which may lead to underfitting) while our techniques select a range of correlations to remove the correlated observations without affecting the performance. The execution time RKPCA based PCA takes 18 seconds, RKPCA based correlation takes 111 seconds, RKPCA ED needs 157 seconds, The RKPCA based cosine technique requires 388 seconds while KPCA demands 417 seconds. The proposed RKPCA are showing great monitoring performance in terms of FAR, MDR, and DTD using all the three fault indices compared with the ordinary KPCA

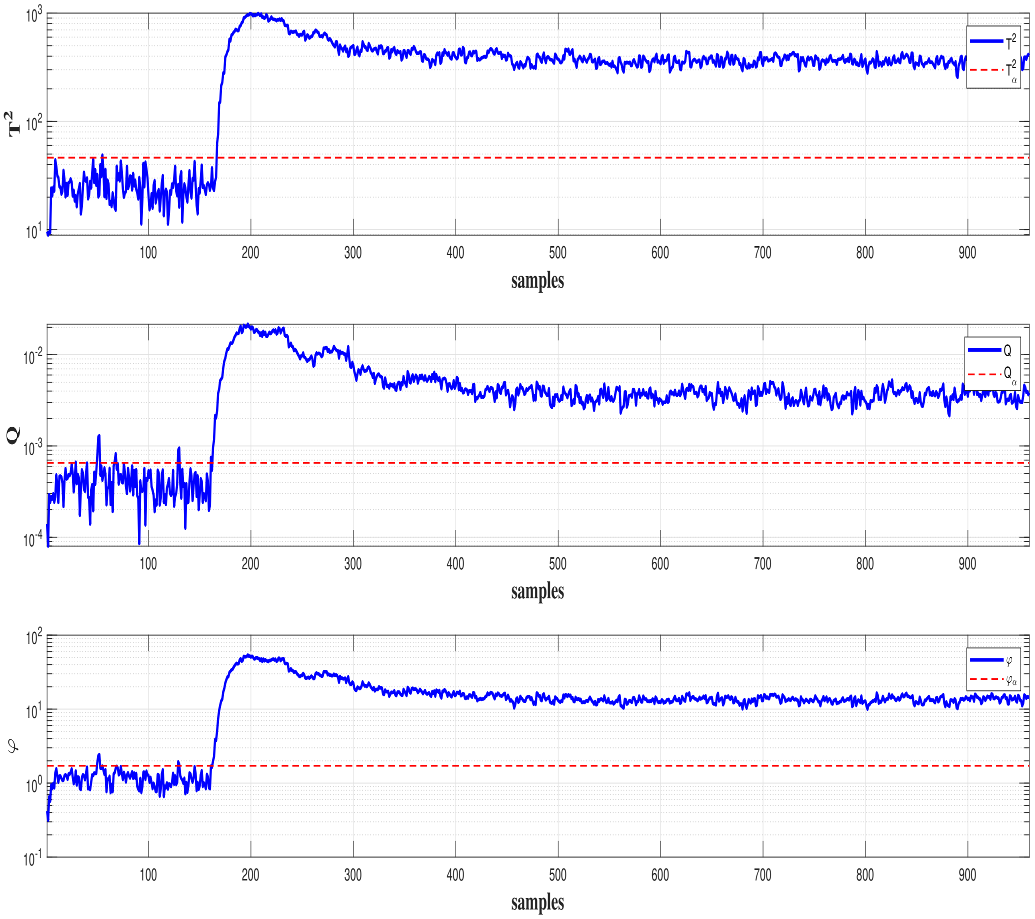


Fig. 5.28 Monitoring results of TEP fault IDV(1) using RKPCA based cosine

and a recently published work RKPCA based cosine [131]. On the other hand, are able to enhance the time and space computations of KPCA algorithm.

Fig. 5.27 reveals the data size reduction and gained computation time percentages using the three proposed RKPCA techniques. RKPCA-ED reduced 71.88% of training observation which gains 62.35% execution time. RKPCA-corr eliminates 78.02% of training observation gaining 73.00% of computation time. Using RKPCA-cosine removes 14.06% of observations and improving the computation time by 6.95%. Fig. 5.28 shows the monitoring plots of fault IDV(1) using RKPCA based cosine. Fig.5.29 plots the monitoring graphs of fault IDV(2) using RKPCA based correlation. Fig. 5.30 presents the monitoring performance of IDV(6) using RKPCA based ED. these are some plots to illustrate the performance of the proposed RKPCA methods in terms of T^2, Q, φ. As time complexity, TEP original dataset has 960observation thus reduce time complexity to $O(N^2)$ it is required to

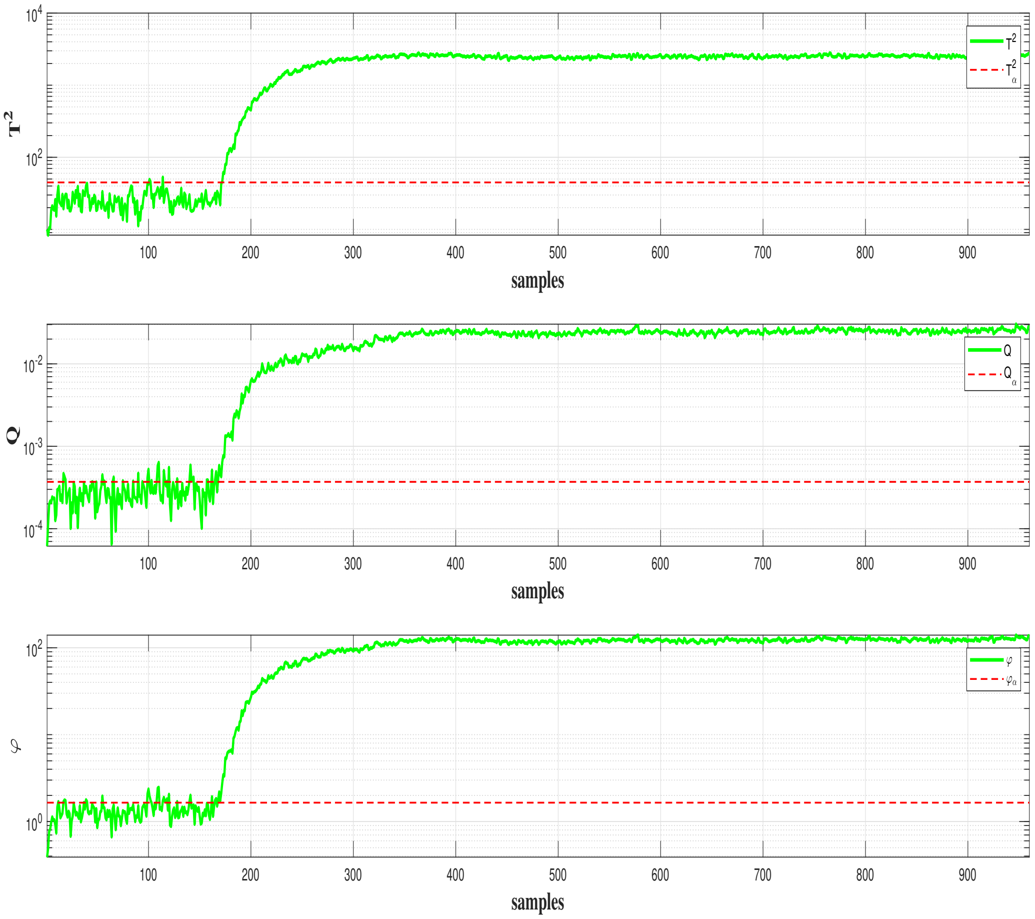


Fig. 5.29 Monitoring results of TEP fault IDV(2) using RKPCA based correlation

reduce the dataset to less than 97. Clearly all the proposed techniques didn't reduce the training data to the required size thus the time complexity is $O(r^3)$.

5.4 Conclusion

The three proposed techniques have been tested on Ain Elkebira and TEP in terms of FAR, MDR, DTD and computation time and memory space, and compared to different FDD techniques. It is clear that the proposed methods have performed well in detecting different faults with small FAR, MDR, DTD, in addition, they have minimizes the time and memory consumption compared to KPCA algorithm. This makes the proposed methods an efficient FDD schemes that can be reliable in monitoring industrial processes.

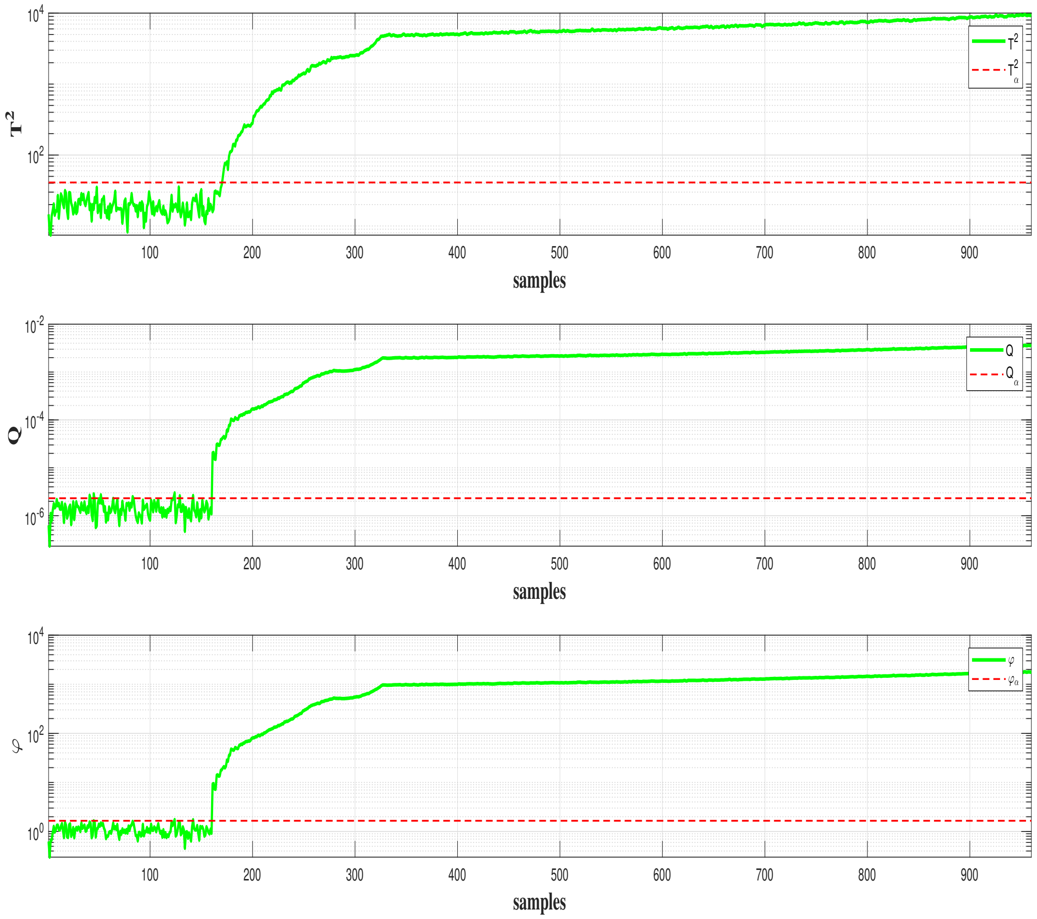


Fig. 5.30 Monitoring results of TEP fault IDV(6) using RKPCA based ED

Table 5.15 The monitoring performance of the different fault indices and $FAR \mid MRD \mid DTD$ via different techniques TEP process

Faults	RKPCA based correlation			RKPCA based PCA [131]			RKPCA based ED [10]		
	T^2	Q	φ	T^2	Q	φ	T^2	Q	φ
IDV(1)	2.50|0.38|2	25.63|0.13|1	22.50|0.00|1	0.00|0.75|5	0.00|0.63|7	0.00|0.38|4	0.00|0.75|7	3.75|0.25|3	1.88|0.25|3
IDV(2)	1.88|1.38|12	17.50|0.50|2	15.00|0.50|2	0.00|1.75|15	0.00|2.13|15	5.00|1.38|12	0.63|1.38|12	3.75|1.50|13	4.38|1.50|13
IDV(3)	11.30|84.38|3	27.5|62.25|1	38.75|49.75|1	0.00|99.88|34	0.00|99.88|32	1.25|92.38|21	0.00|98.75|41	3.75|92.13|3	1.25|93.75|21
IDV(4)	3.13|36.63|1	28.13|0.00|1	26.25|0.00|1	0.00|87.75|1	0.00|49.38|5	1.25|0.13|1	0.63|66.38|1	6.88|0.00|1	1.88|0.00|1
IDV(5)	3.13|63.63|1	28.13|44.75|1	26.25|37.88|1	0.00|78.25|2	0.00|92.38|3	1.25|64.13|1	0.63|75.50|1	6.88|64.75|1	1.88|65.50|1
IDV(6)	0.00|0.50|5	25.00|0.00|1	20.00|0.00|1	0.00|1.38|12	0.00|0.00|1	0.63|0.00|1	0.00|1.00|9	4.38|0.00|1	0.00|0.00|1
IDV(7)	0.63|0.00|1	26.25|0.00|1	23.75|0.00|1	0.00|0.00|1	0.00|0.50|1	0.00|0.00|1	0.00|0.00|1	0.63|0.00|1	0.63|0.00|1
IDV(8)	11.30|1.50|1	36.25|1.00|3	44.38|0.75|3	0.00|3.13|25	0.00|10.38|22	1.25|2.13|16	0.00|2.75|23	4.38|2.25|18	1.88|2.13|18
IDV(9)	20.60|86.25|1	39.38|64.00|1	49.38|51.88|1	0.00|99.63|1	0.00|99.88|121	7.50|93.75|1	1.25|97.75|1	5.00|94.13|1	5.63|94.75|1
IDV(10)	3.13|42.63|8	22.50|16.50|3	20.63|12.3|1	0.00|83.93|55	0.00|98.63|122	1.88|45.38|8	0.00|71.38|28	1.88|54.25|24	1.25|45.63|19
IDV(11)	5.63|36.25|6	25.00|12.25|1	29.38|8.75|1	0.00|63.13|7	0.00|69.50|13	2.50|23.63|6	0.00|54.63|7	2.50|31.00|7	1.88|25.8|6
IDV(12)	1.88|0.88|1	27.50|0.50|1	26.88|0.00|1	0.00|2.25|3	0.00|21.00|4	5.63|1.00|1	0.00|1.50|3	6.25|3.50|3	5.53|0.88|3
IDV(13)	2.50|4.63|8	20.00|3.75|1	14.38|3.63|1	0.00|6.00|47	0.00|5.25|43	3.13|4.50|37	1.25|5.50|38	4.38|4.88|37	2.50|4.63|37
IDV(14)	1.25|0.00|1	26.25|0.00|1	26.88|0.00|1	0.00|2.13|2	0.00|0.38|3	4.38|0.00|1	0.00|0.50|1	6.88|0.00|1	4.38|0.00|1
IDV(15)	33.80|82.13|78	25.00|68.00|1	21.25|55.75|1	0.00|99.88|680	0.00|99.88|677	1.88|92.38|92	0.00|98.75|234	0.00|91.50|133	1.88|92.88|92
IDV(16)	33.80|55.00|1	36.25|19.00|1	50.63|11.75|1	0.00|94.75|291	0.00|98.38|201	10.63|56.63|2	3.75|87.63|12	2.50|55.75|5	4.38|55.50|5
IDV(17)	3.13|12.88|1	31.88|2.88|1	32.50|2.88|1	0.00|27.88|29	0.00|15.38|24	3.13|5.63|20	0.63|22.38|29	7.50|4.75|20	2.50|5.13|20
IDV(18)	1.88|9.13|15	23.12|6.88|1	26.88|5.25|1	0.00|11.38|89	0.00|10.50|85	3.13|9.25|18	0.63|10.75|55	6.25|9.38|18	4.38|9.38|18
IDV(19)	1.25|81.37|11	23.13|28.38|2	18.13|22.75|1	0.00|99.13|78	0.00|99.75|501	4.38|71.13|11	0.00|91.25|11	6.25|78.75|2	2.50|76.00|11
IDV(20)	0.63|40.75|12	20.63|22.25|1	13.75|13.88|1	0.00|82.13|90	0.00|65.25|91	3.13|40.75|68	1.25|82.25|87	3.75|43.38|4	1.25|40.63|11
IDV(21)	8.13|50.50|104	40.63|61.63|1	48.13|30.25|1	0.00|72.25|286	0.00|65.38|370	6.88|49.13|14	0.00|61.38|285	11.88|47.75|2	9.38|48.00|2
J	**3.11**	3.09	3.23	6.01	6.90	2.83	4.40	2.95	2.75

<table>
<tr><th colspan="3">RKPCA Based cosine</th><th colspan="3">KPCA</th></tr>
<tr><th>T^2</th><th>Q</th><th>φ</th><th>T^2</th><th>Q</th><th>φ</th></tr>
<tr><td>0.62|0.75|7</td><td>5.00|0.12|1</td><td>3.12|0.25|3</td><td>0.63|0.75|7</td><td>4.38|0.13|1</td><td>2.50|0.25|3</td></tr>
<tr><td>1.25|1.37|12</td><td>3.12|1.37|2</td><td>2.50|1.50|13</td><td>1.25|1.38|12</td><td>6.88|1.38|2</td><td>3.13|1.50|13</td></tr>
<tr><td>0.62|98.25|21</td><td>4.37|94.12|41</td><td>0.00|92.75|21</td><td>0.63|98.38|21</td><td>6.88|90.13|43</td><td>1.88|93.00|41</td></tr>
<tr><td>1.87|60.87|1</td><td>5.62 |0.00|1</td><td>1.87|0.00|1</td><td>1.25|63.38|1</td><td>5.00|0.00|1</td><td>1.25|0.00|1</td></tr>
<tr><td>1.87|75.00|1</td><td>5.62|63.87|1</td><td>1.87|64.75|1</td><td>1.25|75.13|1</td><td>5.00|61.88|1</td><td>1.88|64.88|1</td></tr>
<tr><td>0.00|1.00|1</td><td>2.50|0.00|1</td><td>0.00|0.00|1</td><td>0.00|1.13|10</td><td>5.63|0.00|1</td><td>1.88|0.00|1</td></tr>
<tr><td>0.00|0.00|1</td><td>0.62|0.00|1</td><td>0.00|0.00|1</td><td>0.00|0.00|1</td><td>2.50|0.00|1</td><td>0.00|0.00|1</td></tr>
<tr><td>0.63|2.75|15</td><td>7.50|2.37|16</td><td>3.75|2.12|20</td><td>0.00|2.75|23</td><td>6.25|2.25|18</td><td>1.88|2.13|18</td></tr>
<tr><td>1.87|98.00|1</td><td>3.75|94.37|1</td><td>6.25|94.50|1</td><td>3.13|97.38|1</td><td>8.75|92.75|1</td><td>6.88|94.75|1</td></tr>
<tr><td>0.00|68.12|8</td><td>1.87|48.25|8</td><td>1.87|41.37|15</td><td>0.00|70.63|28</td><td>3.75|49.50|11</td><td>1.88|44.25|19</td></tr>
<tr><td>0.62|49.50|6</td><td>7.50|32.75|6</td><td>4.37|24.00|6</td><td>0.63|52.50|6</td><td>6.25|27.25|7</td><td>3.75|23.75|6</td></tr>
<tr><td>0.62|1.50|1</td><td>7.50|2.00|3</td><td>5.00|1.00|1</td><td>0.63|1.63|3</td><td>8.75|3.13|3</td><td>5.63|0.88|3</td></tr>
<tr><td>0.62|5.37|36</td><td>5.00|4.37|17</td><td>2.50|4.50|37</td><td>1.25|5.38|37</td><td>4.38|5.00|41</td><td>3.13|4.63|37</td></tr>
<tr><td>0.62|1.25|1</td><td>6.25|0.12|1</td><td>3.12|0.00|1</td><td>0.00|0.50|1</td><td>6.88|0.00|1</td><td>3.13|0.00|1</td></tr>
<tr><td>0.00|98.00|227</td><td>5.00|91.87|88</td><td>1.87|92.25|88</td><td>0.63|98.50|114</td><td>3.13|88.63|66</td><td>1.88|92.63|101</td></tr>
<tr><td>4.37|85.12|12</td><td>3.75|50.37|5</td><td>7.50|47.87|2</td><td>4.38|87.75|149</td><td>5.00|51.50|5</td><td>4.38|53.63|5</td></tr>
<tr><td>0.00|23.12|18</td><td>5.62|4.50|18</td><td>3.75|5.13|18</td><td>0.63|22.00|29</td><td>10.63|4.00|20</td><td>10.63|4.63|20</td></tr>
<tr><td>0.62|70.75|18</td><td>3.75|9.37|18</td><td>3.75|9.25|18</td><td>0.63|10.63|55</td><td>10.00|8.88|16</td><td>5.00|9.25|18</td></tr>
<tr><td>0.00|90.75|2</td><td>6.25|78.37|2</td><td>4.37|74.62|2</td><td>0.00|89.25|11</td><td>8.13|76.63|2</td><td>3.13|73.88|11</td></tr>
<tr><td>0.00|72.62|4</td><td>3.75|43.37|7</td><td>1.87|41.50|7</td><td>0.00|71.38|87</td><td>4.38|42.25|7</td><td>3.13|40.50|12</td></tr>
<tr><td>0.00|58.25|174</td><td>7.5|47.75|1</td><td>5.63|46.62|1</td><td>0.00|60.00|174</td><td>16.25|43.75|1</td><td>6.88|47.00|2</td></tr>
<tr><td>4.01</td><td><b>2.83</b></td><td><b>2.63</b></td><td>4.17</td><td>2.90</td><td>2.80</td></tr>
</table>

6 General conclusion

6.1 General conclusion

New RKPCA methods have been proposed in this dissertation as a solution to the conventional KPCA high time and space computations. The new methods aim to reduce the training dataset size based on some criteria in order to eliminate irrelevant training observations and keep only the relevant samples that can monitor the process with the same quality and less memory and time consumption. The first proposed method reduces the redundant observations among the training dataset based on the Euclidean distance between training samples such that any two samples with zero Euclidean distance are considered similar so that one of these samples can be eliminated afterwards a reduced training dataset. The second method considered the correlated observations as repeated ones as consequence it reserves the uncorrelated observations to build an uncorrelated reduced training dataset. In the third method, The reduction is based on the statistical dependency between the observations. The dependent observations are removed from the dataset. All three methods build a reduced training dataset on which KPCA algorithm is applied to monitor the process with less time and memory consumption. The three techniques have been successfully able to reduce the computational and space costs of ordinary KPCA.

The proposed methods have been utilized to monitor industrial processes: Ain Elkebira cement rotary Kiln and TEP. The results are recorded in terms of FAR, MDR, and DTD contributed by different fault indices: T^2, Q, and φ. A multi-objective index J has been proposed to summarize all the results of different faults indices in one index. The memory and space consumption computed. The proposed RKPCA methods results are compared to other RKPCA techniques, KPCA, and PCA to evaluate the efficiency of these methods. The obtained results show great monitoring results with a good reduction of FAR, MDR, and DTD in addition to less computation time and space are required to perform the process monitoring compared to conventional Kernel PCA.

Although, the proposed RKPCA techniques have preform well in process monitoring and reduce the time and space consumption of the ordinary KPCA. They still struggle to obtain the best similarity, correlation, or Independence threshold that is used to build the reduced training dataset. The convergence of the objective function is not always guaranteed.

6.2 Future work

The proposed techniques are able to solve the KPCA high computational time and space complexity. The performance of the proposed RKPCA techniques could be developed to monitor more complex industrial processes. As future works, it will involve developing extensions of RKPCA methods to monitor dynamic processes. The extensions will include reduced dynamic KPCA (RDKPCA), reduced interval-valued KPCA (RIKPCA), reduced interval-valued KPCA (RIDKPCA), and many others.... The selection of thresholds (similarity, correlation, and Independence) is very important in RKPCA so fast methods to select these thresholds can be developed. Dynamic thresholds can be used to fault indices (T^2, Q, and φ), the dynamic thresholds can minimize FAR and MDR.

The reduction techniques could be used to reduce other kernel features extraction methods like KPLS, kernel Fisher discriminant analysis (KFD), kernel canonical correlation analysis (KCCA) and their different extensions.

References

[1] J. Gertler, *Fault detection and diagnosis in engineering systems*. Routledge, 2017.

[2] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, “A review of process fault detection and diagnosis part II: Qualitative models and search strategies,” *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 313–326, 2003.

[3] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, “A review of process fault detection and diagnosis part III: Process history based methods,” *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 327–346, 2003.

[4] L. Summerer, J. Keller, and M. Darouach, “Robust fault diagnosis with a two-stage kalman estimator,” *European Journal of Control*, vol. 3, no. 3, pp. 247 – 252, 1997.

[5] H. Kazemi and A. Yazdizadeh, “Fault detection and isolation of gas turbine engine using inversion-based and optimal state observers,” *European Journal of Control*, 2020.

[6] X. Wei and M. Verhaegen, “Robust fault detection observer and fault estimation filter design for lti systems based on gkyp lemma,” *European Journal of Control*, vol. 16, no. 4, pp. 366 – 383, 2010.

[7] E. L. Russell, L. H. Chiang, and R. D. Braatz, “Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 51, no. 1, pp. 81–93, 2000.

[8] X. Wei, M. Verhaegen, and T. Van Engelen, “Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques,” *International Journal of Adaptive Control and Signal Processing*, 2010.

[9] H. Niemann, A. Saberi, A. A. Stoorvogel, and P. Sannuti, “Exact, almost and delayed fault detection: An observer based approach,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 9, no. 4, pp. 215–238, 1999.

[10] F. Bencheikh, M. Harkat, A. Kouadri, and A. Bensmail, “New reduced kernel pca for fault detection and diagnosis in cement rotary kiln,” *Chemometrics and Intelligent Laboratory Systems*, vol. 204, p. 104091, 2020.

[11] I. T. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. New York: Springer-Verlag, 2002, vol. 98. [Online]. Available: http://link.springer.com/10.1007/b98835

[12] R. Dunia and S. J. Qin, “Joint diagnosis of process and sensor faults using principal component analysis,” *Control Eng. Pract.*, vol. 6, no. 4, pp. 457–469, 1998.

[13] K. Peason, “On lines and planes of closest fit to systems of point in space,” *Philosophical Magazine*, vol. 2, no. 11, pp. 559–572, 1901.

[14] H. Hotelling, “Analysis of a complex of statistical variables into principal components.” *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[15] J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, “Nonlinear process monitoring using kernel principal component analysis,” *Chemical Engineering Science*, vol. 59, no. 1, pp. 223–234, jan 2004.

[16] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[17] S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, “Fault detection and identification of nonlinear processes based on kernel pca,” *Chemometrics and intelligent laboratory systems*, vol. 75, no. 1, pp. 55–67, 2005.

[18] K. E. Pilario, M. Shafiee, Y. Cao, L. Lao, and S.-H. Yang, “A review of kernel methods for feature extraction in nonlinear process monitoring,” *Processes*, vol. 8, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2227-9717/8/1/24

[19] L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou, “Stochastic optimization for kernel pca,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16. AAAI Press, 2016, p. 2316–2322.

[20] I. Jaffel, O. Taouali, M. F. Harkat, and H. Messaoud, "Moving window kpca with reduced complexity for nonlinear dynamic process monitoring," *ISA Transactions*, vol. 64, pp. 184 – 192, 2016.

[21] A. Perez and C. Felipe, "Fault diagnosis with reconstruction-based contributions for statistical process monitoring," Ph.D. dissertation, Viterbi School of Engineering, 2011.

[22] R. Rosipal and M. Girolami, "An expectation-maximization approach to nonlinear component analysis," *Neural Computation*, vol. 13, no. 3, pp. 505–510, 2001.

[23] Kwang In Kim, M. O. Franz, and B. Scholkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1351–1366, Sep. 2005.

[24] W. Zheng, C. Zou, and L. Zhao, "An improved algorithm for kernel principal component analysis," *Neural Processing Letters*, vol. 22, no. 1, pp. 49–56, aug 2005.

[25] K. Chitta, J. M. Alvarez, E. Haussmann, and C. Farabet, "Less is more: An exploration of data redundancy with active dataset subsampling," *arXiv preprint arXiv:1905.12737*, 2019.

[26] A. Hamadouche, A. Kouadri, and A. Bensmail, "Kernelized relative entropy for direct fault detection in industrial rotary kilns," *International Journal of Adaptive Control and Signal Processing*, apr 2018.

[27] F. I. Khan and S. Abbasi, "Major accidents in process industries and an analysis of causes and consequences," *Journal of Loss Prevention in the process Industries*, vol. 12, no. 5, pp. 361–378, 1999.

[28] V. R. Dhara and R. Dhara, "The union carbide disaster in bhopal: a review of health effects," *Archives of Environmental Health: An International Journal*, vol. 57, no. 5, pp. 391–404, 2002.

[29] B. Bowonder and H. A. Linstone, "Notes on the bhopal accident: Risk analysis and multiple perspectives," *Technological Forecasting and Social Change*, vol. 32, no. 2, pp. 183–202, 1987.

[30] L. W. D. Cullen, "The public inquiry into the piper alpha disaster," *Drilling Contractor;(United States)*, vol. 49, no. 4, 1993.

[31] M. E. Paté-Cornell, “Learning from the piper alpha accident: A postmortem analysis of technical and organizational factors,” *Risk Analysis*, vol. 13, no. 2, pp. 215–232, 1993.

[32] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, “A review of process fault detection and diagnosis: Part i: Quantitative model-based methods,” *Computers and chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[33] M. Mansouri, M.-F. Harkat, H. N. Nounou, and M. N. Nounou, *Data-Driven and Model-Based Methods for Fault Detection and Diagnosis*. Elsevier, 2020.

[34] J. D. Stefanovski, “Passive fault tolerant perfect tracking with additive faults,” *Automatica*, vol. 87, pp. 432–436, 2018.

[35] M. Basseville and Q. Zhang, “Handling parametric and non-parametric additive faults in ltv systems,” *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 523–528, 2015, 9th IFAC Symposium on Fault Detection, Supervision andSafety for Technical Processes SAFEPROCESS 2015.

[36] D. Rotondo, F.-R. López-Estrada, F. Nejjari, J.-C. Ponsart, D. Theilliol, and V. Puig, “Actuator multiplicative fault estimation in discrete-time lpv systems using switched observers,” *Journal of the Franklin Institute*, vol. 353, no. 13, pp. 3176–3191, 2016.

[37] P. Gil, F. Santos, L. Palma, and A. Cardoso, “Recursive subspace system identification for parametric fault detection in nonlinear systems,” *Applied Soft Computing*, vol. 37, pp. 444–455, 2015.

[38] Y. Wu, Y. Wang, Y. Jiang, and Q. Sun, “Multiple parametric faults diagnosis for power electronic circuits based on hybrid bond graph and genetic algorithm,” *Measurement*, vol. 92, pp. 365–381, 2016.

[39] J. J. Gertler, *Fault detection and diagnosis in engineering systems*. CRC Press, 2017.

[40] P. A. Samara, G. N. Fouskitakis, J. S. Sakellariou, and S. D. Fassois, “A statistical method for the detection of sensor abrupt faults in aircraft control systems,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 789–798, 2008.

[41] H. Q. Zhang and Y. Yan, “A wavelet-based approach to abrupt fault detection and diagnosis of sensors,” *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 5, pp. 1389–1396, 2001.

[42] B. Huang, “Detection of abrupt changes of total least squares models and application in fault detection,” *IEEE Transactions on Control Systems Technology*, vol. 9, no. 2, pp. 357–367, 2001.

[43] H. Ji, X. He, J. Shang, and D. Zhou, “Incipient fault detection with smoothing techniques in statistical process monitoring,” *Control Engineering Practice*, vol. 62, pp. 11–21, 2017.

[44] K. Zhang, B. Jiang, X.-G. Yan, and Z. Mao, “Incipient voltage sensor fault isolation for rectifier in railway electrical traction systems,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6763–6774, 2017.

[45] X. Zhang, M. Polycarpou, and T. Parisini, “A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 576–593, 2002.

[46] N. Haje Obeid, A. Battiston, T. Boileau, and B. Nahid-Mobarakeh, “Early intermittent interturn fault detection and localization for a permanent magnet synchronous motor of electrical vehicles using wavelet transform,” *IEEE Transactions on Transportation Electrification*, vol. 3, no. 3, pp. 694–702, 2017.

[47] L. K. Carvalho, J. C. Basilio, M. V. Moreira, and L. B. Clavijo, “Diagnosability of intermittent sensor faults in discrete event systems,” in *2013 American Control Conference*, 2013, pp. 929–934.

[48] W. A. Syed, S. Perinpanayagam, M. Samie, and I. Jennions, “A novel intermittent fault detection algorithm and health monitoring for electronic interconnections,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 6, no. 3, pp. 400–406, 2016.

[49] Q. Yang, “Model-based and data driven fault diagnosis methods with applications to process monitoring,” Ph.D. dissertation, Case Western Reserve University, 2004.

[50] C. KAVURI, “Data-based modeling: Application in process identification, monitoring and fault detection,” Ph.D. dissertation, NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA, 2011.

[51] M. Kinnaert, “Fault diagnosis based on analytical models for linear and nonlinear systems-a tutorial,” *IFAC Proceedings Volumes*, vol. 36, no. 5, pp. 37–50, 2003.

[52] M. Nyberg and C. M. Nyberg, “Model based fault diagnosis: Methods, theory, and automotive engine applications,phdthesis,” Ph.D. dissertation, 1999.

[53] R. N. Clark, D. C. Fosth, and V. M. Walton, “Detecting instrument malfunctions in control systems,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 465–473, 1975.

[54] P. M. Frank, “Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results,” *automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[55] X. Ding and P. M. Frank, “Fault detection via factorization approach,” *Systems and control letters*, vol. 14, no. 5, pp. 431–436, 1990.

[56] R. J. Patton and J. Chen, “A review of parity space approaches to fault diagnosis,” *IFAC Proceedings Volumes*, vol. 24, no. 6, pp. 65–81, 1991.

[57] E. Chow and A. Willsky, “Analytical redundancy and the design of robust failure detection systems,” *IEEE Transactions on automatic control*, vol. 29, no. 7, pp. 603–614, 1984.

[58] K. Benothman, D. Maquin, J. Ragot, and M. Benrejeb, “Diagnosis of uncertain linear systems: an interval approach,” *International Journal of Sciences and Techniques of Automatic control and computer engineering*, vol. 1, no. 2, pp. 136–154, 2007.

[59] Y. Wang and C. Zhao, “Probabilistic fault diagnosis method based on the combination of nest-loop fisher discriminant analysis and analysis of relative changes,” *Control Engineering Practice*, vol. 68, pp. 32–45, 2017.

[60] J. Feng, J. Wang, H. Zhang, and Z. Han, “Fault diagnosis method of joint fisher discriminant analysis based on the local and global manifold learning and its kernel version,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 122–133, 2016.

[61] J. Zheng, H. Wang, Z. Song, and Z. Ge, “Ensemble semi-supervised fisher discriminant analysis model for fault classification in industrial processes,” *ISA Transactions*, vol. 92, pp. 109–117, 2019.

[62] M. Van and H.-J. Kang, “Bearing defect classification based on individual wavelet local fisher discriminant analysis with particle swarm optimization,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 124–135, 2016.

[63] B. Jiang, X. Zhu, D. Huang, J. A. Paulson, and R. D. Braatz, “A combined canonical variate analysis and fisher discriminant analysis (cva–fda) approach for fault diagnosis,” *Computers and Chemical Engineering*, vol. 77, pp. 1–9, 2015.

[64] X. Li, F. Duan, P. Loukopoulos, I. Bennett, and D. Mba, “Canonical variable analysis and long short-term memory for fault diagnosis and performance estimation of a centrifugal compressor,” *Control Engineering Practice*, vol. 72, pp. 177–191, 2018.

[65] P. Odiowei and Y. Cao, “Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations,” in *10th International Symposium on Process Systems Engineering: Part A*, ser. Computer Aided Chemical Engineering, R. M. de Brito Alves, C. A. O. do Nascimento, and E. C. Biscaia, Eds. Elsevier, 2009, vol. 27, pp. 1557–1562.

[66] B. Jiang and R. D. Braatz, “Fault detection of process correlation structure using canonical variate analysis-based correlation features,” *Journal of Process Control*, vol. 58, pp. 131–138, 2017.

[67] F. Guo, C. Shang, B. Huang, K. Wang, F. Yang, and D. Huang, “Monitoring of operating point and process dynamics via probabilistic slow feature analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 151, pp. 115–125, 2016.

[68] J. Huang, O. K. Ersoy, and X. Yan, “Slow feature analysis based on online feature reordering and feature selection for dynamic chemical process monitoring,” *Chemometrics and Intelligent Laboratory Systems*, vol. 169, pp. 1–11, 2017.

[69] N. Zhang, X. Tian, L. Cai, and X. Deng, “Process fault detection based on dynamic kernel slow feature analysis,” *Computers and Electrical Engineering*, vol. 41, pp. 9–17, 2015.

[70] C. Shang, B. Huang, F. Yang, and D. Huang, “Slow feature analysis for monitoring and diagnosis of control performance,” *Journal of Process Control*, vol. 39, pp. 21–34, 2016.

[71] A. Beghi, R. Brignoli, L. Cecchinato, G. Menegazzo, M. Rampazzo, and F. Simmini, “Data-driven fault detection and diagnosis for hvac water chillers,” *Control Engineering Practice*, vol. 53, pp. 79–91, 2016.

[72] H. Jiang, J. J. Zhang, W. Gao, and Z. Wu, “Fault detection, identification, and location in smart grid based on data-driven computational methods,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2947–2956, 2014.

[73] Z. Li, H. Fang, M. Huang, Y. Wei, and L. Zhang, "Data-driven bearing fault identification using improved hidden markov model and self-organizing map," *Computers and Industrial Engineering*, vol. 116, pp. 37–46, 2018.

[74] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun, "Monitoring a robot swarm using a data-driven fault detection approach," *Robotics and Autonomous Systems*, vol. 97, pp. 193–203, 2017.

[75] I. V. de Bessa, R. M. Palhares, M. F. S. V. D'Angelo, and J. E. Chaves Filho, "Data-driven fault detection and isolation scheme for a wind turbine benchmark," *Renewable Energy*, vol. 87, no. P1, pp. 634–645, 2016.

[76] Q. Jiang, S. X. 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 Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8148–8157, 2017.

[77] N. Sadati, R. B. Chinnam, and M. Z. Nezhad, "Observational data-driven modeling and optimization of manufacturing processes," *Expert Systems with Applications*, vol. 93, pp. 456–464, 2018.

[78] E. Naderi and K. Khorasani, "Data-driven fault detection, isolation and estimation of aircraft gas turbine engine actuator and sensors," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–6.

[79] M. Baptista, I. P. de Medeiros, J. P. Malere, C. Nascimento, H. Prendinger, and E. M. Henriques, "Comparative case study of life usage and data-driven prognostics techniques using aircraft fault messages," *Computers in Industry*, vol. 86, pp. 1–14, 2017.

[80] F. Jamil, M. Abid, I. Haq, A. Q. Khan, and M. Iqbal, "Fault diagnosis of pakistan research reactor-2 with data-driven techniques," *Annals of Nuclear Energy*, vol. 90, pp. 433–440, 2016.

[81] X. Li, N. Wang, L. Wang, I. Kantor, J.-L. Robineau, Y. Yang, and F. Maréchal, "A data-driven model for the air-cooling condenser of thermal power plants based on data reconciliation and support vector regression," *Applied Thermal Engineering*, vol. 129, pp. 1496–1507, 2018.

[82] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, feb 1991.

[83] D. Dong and T. J. McAvoy, "Nonlinear principal component analysis—based on principal curves and neural networks," *Computers and Chemical Engineering*, vol. 20, no. 1, pp. 65–78, 1996.

[84] S. Qin and T. McAvoy, "Nonlinear pls modeling using neural networks," *Computers and Chemical Engineering*, vol. 16, no. 4, pp. 379–391, 1992, neutral network applications in chemical engineering.

[85] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[86] F. Harrou, M. N. Nounou, H. N. Nounou, and M. Madakyaru, "Statistical fault detection using pca-based glr hypothesis testing," *Journal of loss prevention in the process industries*, vol. 26, no. 1, pp. 129–139, 2013.

[87] S. J. Qin, "Statistical process monitoring: basics and beyond," *Journal of Chemometrics*, vol. 17, no. 8-9, pp. 480–502, 2003.

[88] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[89] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive pca for adaptive process monitoring," *Journal of process control*, vol. 10, no. 5, pp. 471–486, 2000.

[90] W. Liang and L. Zhang, "A wave change analysis (wca) method for pipeline leak detection using gaussian mixture model," *Journal of Loss Prevention in the Process Industries*, vol. 25, no. 1, pp. 60–69, 2012.

[91] B. R. Bakshi, "Multiscale pca with application to multivariate statistical process monitoring," *AIChE journal*, vol. 44, no. 7, pp. 1596–1610, 1998.

[92] J.-C. Jeng, "Adaptive process monitoring using efficient recursive pca and moving window pca algorithms," *Journal of the Taiwan Institute of Chemical Engineers*, vol. 41, no. 4, pp. 475–481, 2010.

[93] P. Nomikos and J. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, no. 8, pp. 1361–1375, 1994.

[94] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 30, no. 1, pp. 179–196, 1995.

[95] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[96] M.-F. Harkat, S. Djelel, N. Doghmane, and M. Benouaret, “Sensor fault detection, isolation and reconstruction using nonlinear principal component analysis,” *International Journal of Automation and Computing*, vol. 4, no. 2, pp. 149–155, 2007.

[97] I. T. Jolliffe, “Discarding variables in a principal component analysis. i: Artificial data,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 21, no. 2, pp. 160–173, 1972.

[98] R. B. Cattell, “The scree test for the number of factors,” *Multivariate Behavioral Research*, vol. 1, no. 2, pp. 245–276, 1966, pMID: 26828106.

[99] W. R. Zwick and W. F. Velicer, “Comparison of five rules for determining the number of components to retain.” *Psychological bulletin*, vol. 99, no. 3, p. 432, 1986.

[100] R. Ledesma and P. Valero-Mora, “Determining the number of factors to retain in efa: an easy-to- use computer program for carrying out parallel analysis,” *Practical Assessment, Research and Evaluation*, vol. 12, 08 2007.

[101] H. H. Yue and S. J. Qin, “Reconstruction-based fault identification using a combined index,” *Industrial and Engineering Chemistry Research*, vol. 40, no. 20, pp. 4403–4414, 2001. [Online]. Available: https://doi.org/10.1021/ie000141+

[102] C. Cheng and M.-S. Chiu, “Nonlinear process monitoring using jitl-pca,” *Chemometrics and Intelligent Laboratory Systems*, vol. 76, no. 1, pp. 1–13, 2005.

[103] H. G. Hiden, M. J. Willis, M. Tham, P. Turner, and G. A. Montague, “Non-linear principal components analysis using genetic programming,” *Computers and Chemical Engineering*, 1997.

[104] F. Jia, E. B. Martin, and A. J. Morris, “Non-linear principal components analysis with application to process fault detection,” *International Journal of Systems Science*, vol. 31, no. 11, pp. 1473–1487, 2000.

[105] U. Kruger, D. Antory, J. Hahn, G. W. Irwin, and G. McCullough, “Introduction of a nonlinearity measure for principal component models,” *Computers and chemical engineering*, vol. 29, no. 11-12, pp. 2355–2362, 2005.

[106] S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 1, pp. 55–67, jan 2005.

[107] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring with kernel principal component analysis," *Industrial and Engineering Chemistry Research*, vol. 49, no. 17, pp. 7849–7857, 2010.

[108] A. Bakdi and A. Kouadri, "An improved plant-wide fault detection scheme based on pca and adaptive threshold for reliable process monitoring: Application on the new revised model of tennessee eastman process," *Journal of Chemometrics*, vol. 32, no. 5, p. e2978, 2018, e2978 CEM-17-0073.R1. [Online]. Available: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.2978

[109] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79 – 86, 1951.

[110] N. Anderson, P. Hall, and D. Titterington, "Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates," *Journal of Multivariate Analysis*, vol. 50, no. 1, pp. 41–54, 1994.

[111] P. A. Olsen and S. Dharanipragada, "An efficient integrated gender detection scheme and time mediated averaging of gender dependent acoustic models," in *8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003, Geneva, Switzerland, September 1-4, 2003*. ISCA, 2003. [Online]. Available: http://www.isca-speech.org/archive/eurospeech_2003/e03_2509.html

[112] J. Silva and S. Narayanan, "Average divergence distance as a statistical discrimination measure for hidden markov models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 890–906, 2006.

[113] T. Zhang, "From ξ-entropy to kl-entropy: Analysis of minimum information complexity density estimation," *The Annals of Statistics*, vol. 34, no. 5, pp. 2180–2210, 2006.

[114] T. van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[115] J. R. Hershey and P. A. Olsen, “Approximating the kullback leibler divergence between gaussian mixture models,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*, vol. 4, 2007, pp. IV–317–IV–320.

[116] F. Perez-Cruz, “Kullback-leibler divergence estimation of continuous distributions,” in *2008 IEEE International Symposium on Information Theory*, 2008, pp. 1666–1670.

[117] B. Scholkopf, “The kernel trick for distances,” *Advances in neural information processing systems*, pp. 301–307, 2001.

[118] M. A. Aizerman, E. A. Braverman, and L. Rozonoer, “Theoretical foundations of the potential function method in pattern recognition learning.” in *Automation and Remote Control,*, no. 25, 1964, pp. 821–837.

[119] C. van den Berg, J. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, ser. Graduate Texts in Mathematics. Springer New York, 2012.

[120] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, 1992, pp. 144–152.

[121] J. Mercer and A. R. Forsyth, “Functions of positive and negative type, and their connection with the theory of integral equations,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 83, no. 559, pp. 69–70, 1909.

[122] R. Hefner, “Warren s. torgerson, theory and methods of scaling. new york: John wiley and sons, inc., 1958. pp. 460,” *Behavioral Science*, vol. 4, no. 3, pp. 245–247, 1959.

[123] V. N. Vapnik, “The nature of statistical learning,” *Theory*, 1995.

[124] G. Wahba, *Spline Models for Observational Data*, ser. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.

[125] S. Bergman, *The kernel function and conformal mapping*. American Mathematical Soc., 1970, vol. 5.

[126] C. A. Micchelli, M. Pontil, and P. Bartlett, “Learning the kernel function via regularization.” *Journal of machine learning research*, vol. 6, no. 7, 2005.

[127] S. Han, C. Qubo, and H. Meng, “Parameter selection in svm with rbf kernel function,” in *World Automation Congress 2012*. IEEE, 2012, pp. 1–4.

[128] J. H. Min and Y.-C. Lee, “Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters,” *Expert systems with applications*, vol. 28, no. 4, pp. 603–614, 2005.

[129] H. Lahdhiri, I. Elaissi, O. Taouali, M. F. Harakat, and H. Messaoud, “Nonlinear process monitoring based on new reduced Rank-KPCA method,” *Stochastic Environmental Research and Risk Assessment*, vol. 32, no. 6, pp. 1833–1848, jun 2018.

[130] Y. Rathi, S. Dambreville, and A. Tannenbaum, “Statistical shape analysis using kernel pca,” in *Image processing: algorithms and systems, neural networks, and machine learning*, vol. 6064. International Society for Optics and Photonics, 2006, p. 60641B.

[131] M.-F. Harkat, A. Kouadri, R. Fezai, M. Mansouri, H. Nounou, and M. Nounou, “Machine Learning-Based Reduced Kernel PCA Model for Nonlinear Chemical Process Monitoring,” *Journal of Control, Automation and Electrical Systems*, pp. 1–14, may 2020.

[132] K. Miettinen, *Nonlinear Multiobjective Optimization*, ser. International Series in Operations Research and Management Science. Springer US, 1999. [Online]. Available: https://books.google.dz/books?id=ha_zLdNtXSMC

[133] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making — Methods and Applications: A State-of-the-Art Survey*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979. [Online]. Available: https://doi.org/10.1007/978-3-642-45511-7_3

[134] S. Pllana, S. Memeti, and J. Kolodziej, “Customizing pareto simulated annealing for multi-objective optimization of control cabinet layout,” *CoRR*, vol. abs/1906.04825, 2019. [Online]. Available: http://arxiv.org/abs/1906.04825

[135] H. A. Nguyen, Z. van Iperen, S. Raghunath, D. Abramson, T. Kipouros, and S. Somasekharan, “Multi-objective optimisation in scientific workflow,” *Procedia Computer Science*, vol. 108, pp. 1443–1452, 2017, international

Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917308062

[136] T. Ganesan, I. Elamvazuthi, and P. Vasant, "Multiobjective design optimization of a nano-cmos voltage-controlled oscillator using game theoretic-differential evolution," *Applied Soft Computing*, vol. 32, pp. 293–299, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494615001726

[137] C. Felipe and A. Perez, "FAULT DIAGNOSIS WITH RECONSTRUCTION-BASED by," Ph.D. dissertation, Viterbi School of Engineering, 2011.

[138] A. Bakdi, A. Kouadri, and A. Bensmail, "Fault detection and diagnosis in a cement rotary kiln using PCA with EWMA-based adaptive threshold monitoring scheme," *Control Eng. Pract.*, vol. 66, no. March, pp. 64–75, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2017.06.003

[139] M. Ammiche, A. Kouadri, and A. Bensmail, "A Modified Moving Window dynamic PCA with Fuzzy Logic Filter and application to fault detection," *Chemom. Intell. Lab. Syst.*, vol. 177, pp. 100–113, 2018. [Online]. Available: https://doi.org/10.1016/j.chemolab.2018.04.012

[140] A. Kouadri, A. Bensmail, A. Kheldoun, and L. Refoufi, "An adaptive threshold estimation scheme for abrupt changes detection algorithm in a cement rotary kiln," *Journal of Computational and Applied Mathematics*, vol. 259, pp. 835–842, 2014.

[141] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers and Chemical Engineering*, 1993.

[142] L. H. Chiang, E. L. Russell, and R. D. Braatz, "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, 2000.