N° Order........../IGEE/UMBB/2025

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
M'HAMED BOUGARA UNIVERSITY OF BOUMERDES

Institute of Electrical and Electronic Engineering

PhD Thesis

Presented by:

Mohammed Tahar Habib KAIB

In Partial Fulfillment of the Requirement for the Degree of

DOCTORATE

Field: Automatique
Option: Automatique

Title:

Multivariate Statistical Process Monitoring Using Kernel Statistical Techniques

---

JURY MEMBERS:

Pr.	Abdelhakim	KHOUAS	Prof	UMB-Boumerdes, Algeria	President
Pr.	Mohamed-Faouzi	HARKAT	Prof	ENSTI-Annaba, Algeria	Supervisor
Pr.	Abdelmalek	KOUADRI	Prof	UMB-Boumerdes, Algeria	Co-Supervisor
Pr.	Ahmed	CHAIB	Prof	UMB-Boumerdes, Algeria	Examiner
Pr.	Abderrazak	LACHOURI	Prof	University of Skikda, Algeria	Examiner
Pr.	Majdi	MANSOURI	Prof	Sultan Qaboos University, Oman	Invited

Academic year: 2024/2025

Dedication

This work is dedicated to:

*This piece of work holds deep personal meaning, as it is dedicated to the loving memory of my late **father** and my dear brother, **Bahri**, may Allah grant them peace. My heart also goes out to my truly invaluable mother. And, naturally, to my family, Zakaria, Fatma, and Rafik, and to my friends: Abdelhalim LOUIFI, Abdelmoumen MESSILEM, Abdesamia AZIZI, Idriss BABAGHAYOU, Lokmane DIBES, Mohammed RENANE, and Mohammed SAHARI.*

Acknowledgements

This work would not have been possible without the outstanding support and guidance of my supervisors, Professor Mohamed-Faouzi HARKAT and Professor *Abdelmalek KOUADRI. Their invaluable suggestions truly shaped this research. I also want to* thank Professor *Mansouri MAJDI* and Professor *Vicenç Puig. A special note of thanks goes to the Signals and Systems Research Laboratory* for the foundational support.

Abstract

Fault Detection and Diagnosis (FDD) is an important part of industrial plants because monitoring systems are responsible for capturing faults as soon as they occur to avoid major casualties in equipment, operators, and the environment. FDD can be classified into two major categories: model-based approaches and model-free approaches. For large systems such as industrial plants, it is difficult to build a model-based monitoring system because these plants are complex and difficult to identify. Model-free approaches are more flexible and simpler to develop for these systems since they rely directly on plant data.

Principal Component Analysis (PCA) is a well-known model-free approach used for fault detection due to its efficiency and lower complexity compared to other methods. Unfortunately, PCA is only designed for systems that exhibit linear characteristics between variables. Kernel PCA (KPCA) is one of the alternatives of PCA that were developed to overcome this problem. KPCA's main idea is to map data from the input space to a higher-dimensional space via a kernel function and then apply PCA there. When dealing with large data sets for monitoring, KPCA faces several challenges: increased storage requirements, longer execution times, and potential degradation in monitoring performance. Reduced KPCA (RKPCA) is an alternative to KPCA used to overcome problems related to the size of the data set. RKPCA reduces the number of samples in the training data set while retaining most of the information in the resulting reduced data set, which is then used to build the KPCA monitoring model.

For this thesis, three RKPCA-based algorithms were proposed to reduce the size of the data set without a significant loss of information from the original data set. This study aims to achieve impressive monitoring performances and surpass existing ones. The first proposed algorithm is the Correlation Dimension RKPCA. It uses chaos theory and fractal dimension to select samples that share the same correlation dimension as the original data set. Keep in mind, this approach is only applicable if the system being monitored is chaotic. The second algorithm is the Variogram-based RKPCA; this algorithm uses spatial continuity, specifically a variogram, to retain only non-correlated samples from the original data set. The last algorithm is the Histogram-based RKPCA, which uses class intervals (histograms) to reduce the size of the data set by maintaining the same data distribution as the original.

These three algorithms were thoroughly evaluated using the Tennessee Eastman Process, a well-known simulated chemical plant, and real-world data from the Ain El Kebira cement plant in Algeria. This diverse evaluation not only confirmed their effectiveness but also facilitated direct comparison with established methods. Ultimately, the results obtained from both applications were decent and satisfactory.

Keywords: Fault Detection; Principal Component Analysis (PCA); Kernel Principal Component Analysis (KPCA); Reduced Kernel Principal Component Analysis (RKPCA); Non-linear processes; Correlation Dimension; Variogram; Class Interval; Histogram.

Résumé

La Détection et le Diagnostic des Défauts (FDD) est une partie importante des usines industrielles, car les systèmes de surveillance sont responsables de capturer les défauts dès qu'ils se produisent afin d'éviter des accidents majeurs sur les équipements, les opérateurs et l'environnement. FDI se divise en deux grandes familles: les approches basées sur des modèles et celles basées sur les données. Pour les grands systèmes comme les usines industrielles, il est difficile de construire un système de surveillance basé sur des modèles, car ces usines sont complexes et difficiles à identifier. Les approches bassées sur les données, en revanche, sont plus flexibles et plus faciles à construire pour ces systèmes car elles sont basées sur les données collectées de l'usine.

L'Analyse en Composantes Principales (PCA) est une approche bassées sur les données bien connue et utilisée pour la détection de défauts en raison de sa simplicité et de sa moindre complexité par rapport à d'autres méthodes. Malheureusement, PCA est uniquement conçue pour les systèmes ayant des caractéristiques linéaires entre les variables. PCA à Noyau (KPCA) est l'une des alternatives à PCA qui a été développée pour surmonter ce problème. L'idée principale du KPCA est de mapper les données de l'espace d'entrée à un espace de dimension supérieure par une function à noyau, puis d'appliquer le PCA dans cet espace. Si l'ensemble de données utilisé pour le modèle de surveillance est grand, alors la KPCA confronte certains nouveaux défis car elle prendra plus d'espace de stockage pour le modèle, le temps d'exécution sera plus long et dans certains cas, elle peut perdre ses performances de surveillance. PCA à Noyau Réduite (RKPCA) est une alternative à la KPCA utilisée pour surmonter les problèmes liés à la taille de l'ensemble de données. La RKPCA réduit le nombre d'échantillons dans l'ensemble de données d'entraînement et conserve la plupart des informations dans l'ensemble de données réduit résultant qui est ensuite utilisé pour construire le modèle de surveillance KPCA.

Pour cette thèse, trois algorithmes basés sur la RKPCA ont été proposés pour réduire la taille de l'ensemble de données sans une perte importante des informations de l'ensemble de données d'entraînement. Cette étude vise à donner des performances de surveillance impressionnantes et à surmonter les performances existantes. Le premier algorithme proposé est la RKPCA basée sur la Dimension de Corrélation qui utilise la théorie du chaos, la dimension fractale, pour conserver certaines observations avec la même dimension de corrélation que l'ensemble de données original. La RKPCA basée sur la Dimension de Corrélation nécessite que le système surveillé soit un système chaotique pour être appliqué. Le deuxième algorithme est la RKPCA basée sur le Variogramme, cet algorithme utilise la continuité spatiale, le variogramme, pour ne retenir que les échantillons non corrélés de l'ensemble de données original. Le dernier algorithme est la RKPCA basée sur l'Histogramme qui utilise l'intervalle de classe, l'histogramme, pour réduire la taille de l'ensemble de données en conservant la même distribution des données que celle de l'ensemble de données original.

Ces trois algorithmes ont été rigoureusement évalués en utilisant le Tennessee Eastman Process, une simulation d'un processus chimique bien connue, et des données réelles de l'usine de ciment d'Ain El Kebira en Algérie. Cette évaluation diversifiée a non seulement confirmé leur efficacité, mais a également facilité la comparaison directe avec d'autres méthodes. Finalement, les résultats obtenus des deux applications se sont avérés décents et satisfaisants.

Ces trois algorithmes et d'autres sont appliqués aux ensembles de données du Processus Tennessee Eastman et de l'usine de ciment de Ain El Kebira pour tester leurs performances et les comparer avec celles existantes. Les résultats obtenus en utilisant ces approches sont remarquables et satisfaisants.

Mots clés: Détection de Défauts; Analyse en Composantes Principales (PCA); Analyse en Composantes Principales à Noyau (KPCA); Analyse en Composantes Principales à Noyau Réduite(RKPCA); Processus Non-linéaires; Dimension de Corrélation; Variogramme; Intervalle de Classe; Histogramme.

ملخص

اكتشاف الأعطال وتشخيصها هو جزء مهم من المصانع لأن أنظمة المراقبة مسؤولة عن اكتشاف الأعطال فور حدوثها لتجنب خسائر كبيرة في المعدات والعمال والبيئة. و يمكن تصنيفها إلى فئتين رئيسيتين: النهج القائم على النموذج والنهج القائم على البيانات. بالنسبة للأنظمة الكبيرة مثل المصانع يصعب بناء نظام مراقبة قائم على النموذج لأن هذه المصانع معقدة وصعبة التحديد. من ناحية أخرى، النهج القائم على البيانات أكثر مرونة وأسهل في الإنجاز بالنسبة لهذه الأنظمة لأنها تعتمد على البيانات التي يتم جمعها من المصنع.

تحليل المركبات الرئيسية هو نهج معروف قائم على البيانات يستخدم لاكتشاف الأعطال بسبب بساطته وقلة تعقيدً بالنسبة للطرق الأخرى. لسوء الحظ، تم تصميمه فقط للأنظمة التي لها خصائص خطية بين المتغيرات. يعد تحليل المركبات الرئيسية بالنواة أحد بدائل التي تم تطويرها للتغلب على هذه المشكلة. فكرته الرئيسية هي تحويل البيانات من فضاء المدخلات إلى فضاء ذات أبعاد أعلى عبر دالة النواة، ثم تطبيق تحليل المركبات الرئيسية هناك. إذا كان حجم مجموعة البيانات المستخدمة في نموذج المراقبة كبيرًا، فإنه سيواجه بعض التحديات الجديدة، مثل الحاجة إلى مساحة تخزين أكبر للنموذج، وزيادة وقت التنفيذ، وفي بعض الحالات، يمكن أن يفقد أدائه في المراقبة. يعتبر تحليل المركبات الرئيسية بالنواة المقلص بديلاً لتحليل المركبات الرئيسية بالنواة يستخدم للتغلب على المشاكل المتعلقة بحجم مجموعة البيانات. يقوم هذا الأخير بتقليل عدد العينات في مجموعة البيانات التدريبية بحيث يحافظ على معظم المعلومات الموجودة في مجموعة البيانات المقلصة وتستخدم هذه البيانات المقلصة بعد ذلك لبناء نموذج مراقبة.

في هذه الرسالة، تم اقتراح ثلاث خوارزميات تعتمد على تحليل المركبات الرئيسية بالنواة المقلص لتقليل حجم مجموعة البيانات دون فقدان كثير من المعلومات في مجموعة البيانات التدريبية. تهدف هذه الدراسة إلى تحقيق أداء مراقبة متميز و متجاوزا للأنظمة الحالية. الخوارزمية الأولى المقترحة هي تحليل المكونات الرئيسية بالنواة المقلص بناءً على البُعد التوافقي، والتي تستخدم نظرية الفوضى والأبعاد الفراغية للحفاظ على بعض العينات بنفس البُعد التوافقي لمجموعة البيانات الأصلية. يتطلب تحليل المركبات الرئيسية بالنواة المقلص بناءً على البُعد التوافقي أن يكون النظام المرصود نظامًا فوضويًا لتطبيقه. الخوارزمية الثانية هي تحليل المركبات الرئيسية بالنواة المقلص المعتمدة على الاستمرارية المكانية حيث تستخدم هذه الخوارزمية الاستمرارية المكانية للاحتفاظ فقط بالعينات غير المرتبطة من مجموعة البيانات الأصلية. الخوارزمية الأخيرة هي تحليل المركبات الرئيسية بالنواة المقلص المعتمدة على الهايستوغرام والتي تقلل حجم مجموعة البيانات عن طريق

الحفاظ على نفس توزيع البيانات كما في البيانات الأصلية.

تم تقييم هذه الخوارزميات الثلاث بدقة باستخدام بيانات من تينيسي إيستمان، وهي محاكات لمحطة كيميائية معروفة، وبيانات حقيقية من مصنع إسمنت عين الكبيرة في الجزائر. هذا التقييم المتنوع لم يؤكد فعاليتها فحسب، بل سهّل أيضًا المقارنة المباشرة مع الأساليب المقترحة من قبل. في النهاية،كانت النتائج التي تم الحصول عليها من كلا التطبيقين جيدة ومرضية.

الكلمات المفتاحية: اكتشاف الأعطال؛ تحليل المركبات الرئيسية ؛ تحليل المركبات الرئيسية بالنواة ؛ تحليل المركبات الرئيسية بالنواة المقلصة ؛ العمليات غير الخطية؛ بُعد الارتباط؛ الفاريوغرام؛ الهايستوغرام.

Contents

List of Figures

List of Tables

Nomenclature

CD	Correlation Dimension.
C_I	Correlation Integral.
CP	Cement Plant.
CPV	Cumulative Percent Variance.
CUMSUM	Cummulative Sum.
DTD	Detection Time Delay.
EWMA	Exponentialy Weighted Moving Average.
FAR	False Alarm Rate.
FDD	Fault Detection and Diagnosis.
KF	Kalman Filter.
KPCA	Kernel Principal Component Analysis.
LCL	Lower Control Limit.
MDR	Missed Detection Rate.
MSPM	Multivariate Statistical Process Monitoring.
NN	Neural Network.
PC	Principal Components.
PCA	Principal Component Analysis.
PLS	Partial Least Squares.
QTA	Qualitative Trend Analysis.
RBF	Radial Basis Function.
RKPCA	Reduced Kernel Principal Component Analysis.
RPF	Real Process Fault.
SDG	Signed Digraphs
SPE	Squared Prediction Error.
SPM	Statistical Process Monitoring.
TEP	Tennessee Eastman Process.
UCL	Upper Control Limit.

General Introduction

Importance of Fault Detection and Diagnosis

In real-world systems, the constant risk of faults can severely affect plant devices and subsystems, such as sensors and actuators. These issues can disrupt normal operating conditions and even endanger personnel[1, 2]. As a result of rapidly developing technologies, industrial systems have become more complicated and sophisticated than before, so observing these systems all the time is a must. Fault Detection and Diagnosis (FDD) systems are used to fulfil this task because they ensure that monitored systems are in a healthy state to avoid casualties, revealing the importance of these FDD systems. For such an important role, FDD systems must be reliable, accurate, and fast. An example of a tragedy that happened due to a failure of the monitoring system is the Bhopal Gas Tragedy; this incident occurred in 1981 at the Union Carbide India Limited pesticide plant in Bhopal, India. This incident was caused by chemicals, water mixed with methyl isocyanate gas in one tank, which caused an exothermic reaction that resulted in a massive increase in pressure. The consequences of this incident were the following.

- The immediate death toll is estimated to range from 3000 to 5000 people, with long-term estimates reaching up to 25,000.
- Environmental contamination and long-lasting health effects on the local population.
- More than 500,000 people were exposed to the toxic gas, causing severe respiratory problems, eye irritation, and other long-term health problems.

All of this happened because ineffective monitoring systems failed to detect the early signs of the chemical reaction. This incident underscores the critical importance of robust fault detection and diagnosis systems in industrial processes to prevent such devastating accidents.

Motivation

This work utilizes Reduced Kernel Principal Component Analysis (RKPCA) for model-free fault detection. Current RKPCA methods struggle with practical limitations, often requiring either predefined parameters or extensive computations for data set reduction. A key challenge is their potential failure to maintain homogeneity between the reduced and original data sets. Preserving this homogeneity is crucial for ensuring the reduced data accurately represent the original.

Three RKPCA algorithms are introduced to address issues related to managing a large number of observations. Notably, these three algorithms do not require optimization or predefined number of clusters. The first approach exploits chaos theory and utilizes the correlation dimension to retain only the most relevant observations from the original data set. This reduction method is specifically designed to minimize the execution time and memory storage requirements. It is particularly useful in scenarios where the embedded system must perform additional tasks while maintaining the FDD capabilities.

The second algorithm employs a variogram-based approach to select non-correlated samples to form the reduced dataset. This method aims to preserve homogeneity with the original dataset while effectively reducing the number of observations. By maintaining a high level of similarity to the original data, this approach ensures that the fault detection and diagnosis performance remains robust and reliable.

The third and final algorithm utilizes a histogram-based method to select representative observations, forming a reduced dataset that maintains the original distribution. This approach aims to balance the preservation of data homogeneity with reductions in both execution time and memory storage requirements. By retaining a data set that reflects the original distribution, the algorithm achieves a compromise between efficiency and performance, delivering a satisfactory overall monitoring outcome.

Processes Used in the Study

To evaluate algorithms, two distinct data sets are employed: one from the Tennessee Eastman Process (TEP) and the other from the Ain El Kebira Cement Plant (CP) Rotary Kiln. Both data sets exhibit nonlinear characteristics, which provides a robust test of the algorithms' capability to manage nonlinearity. The primary goal is to assess whether the proposed algorithms can effectively handle the nonlinear nature of the data while successfully reducing the number of observations. Additionally, it is crucial to determine if these algorithms preserve the fundamental characteristics of the original data set despite the reduction in data size.

Objectives of the Study

The primary aim of this dissertation is to address and overcome the main limitations of KPCA concerning the size of the training data set. The focus of this study is to develop effective strategies for reducing the number of samples in the training data while maintaining optimal performance. This involves employing various techniques to identify and remove irrelevant or redundant samples from the data set.

The reduced data set should achieve several key objectives:

- *Decrease execution time:* The reduction should streamline the online phase of the KPCA algorithm, leading to faster processing times.
- *Minimize Storage Requirements:* The size of the monitoring model should be reduced, lowering the storage space needed.
- *Maintain High Monitoring Performance:* The reduced data set should deliver monitoring performance comparable to or better than that of the conventional KPCA.

Additionally, the reduced data must preserve its homogeneity with the original data set to ensure that it can credibly replace the original data without compromising the integrity of the monitoring results. The proposed reduction methods are designed to be relatively fast and efficient, ensuring that they do not require excessive computational time.

Layout of this Thesis

This thesis is structured as follows:

- **Chapter 1:** This chapter introduces the fundamental concepts of FDD. It covers essential knowledge about various fault detection approaches, Multivariate Statistical Process Monitoring, and introduces key FDD metrics.
- **Chapter 2:** This chapter presents literature review, PCA, and KPCA within the context of Multivariate Statistical Process Monitoring. It delves into the mathematical foundations of these techniques as applied to fault detection.
- **Chapter 3:** This chapter provides a detailed explanation of the proposed RKPCA algorithms and the homogeneity test employed in this study. It also presents a review of related work in the field.
- **Chapter 4:** This chapter details the application of the proposed algorithms to both the Tennessee Eastman process and the Ain El Kebira cement plant, following their introduction. A comparison of the proposed method's performance against existing techniques is presented, evaluating the effectiveness of the monitoring, execution time, required storage space, and homogeneity with the original dataset. The obtained results are discussed in detail.
- Comprehensive Conclusion of the work, summarizing the findings and contributions of the thesis. This chapter also gives a hint of future work.

1 FDD Generalities and Background

1.1 Introduction

In recent years, FDD approaches have seen significant advancements due to their critical importance in various industries. Researchers have been actively working on both developing new FDD methodologies and improving the performance of existing ones. This chapter aims to provide foundational knowledge about faults and FDD systems, offering an overview of the essential attributes that make an FDD system effective. It discusses the fundamental concepts of faults and the various approaches to FDD, highlighting the key features and capabilities that are desirable in an FDD system. By covering different FDD strategies, this chapter helps readers gain a comprehensive understanding of the field, including how different approaches can be applied and what criteria are important for evaluating their effectiveness.

1.2 Faults, Failures, and Malfunctions

The nature of the operating condition of a given process can be either healthy or faulty, a faulty operating condition means that this system does not work properly, and it contains one or more faults. A fault can be classified as a malfunction or as a complete failure. Figure 1 is utilised to distinguish between a failure and a malfunction. The plots presented at the bottom of this figure illustrate the distinct behaviours of these two fault types over time. In both plots, the y-axis represents the fault status, with a value of 0 indicating normal operation (no fault) and 1 signifying the presence of a fault. The x-axis for both plots consistently denotes the progression of time.

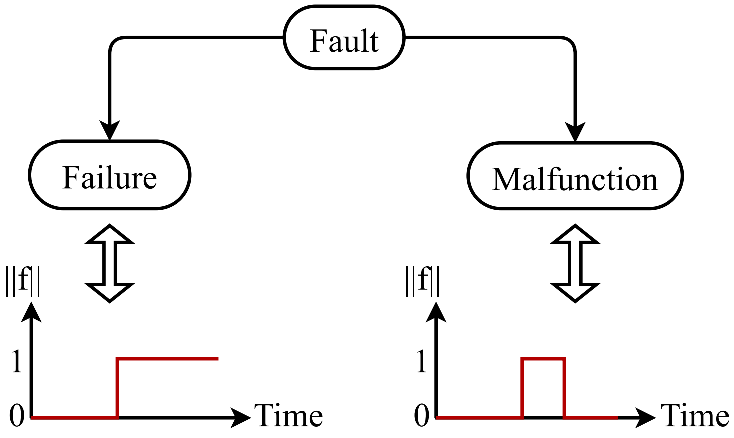

Figure 1: Distinguishing Between Failure and Malfunction via Fault Status.

To clarify, let's define fault, malfunction, and failure in this context:

1. Fault: A fault is an unacceptable deviation of one variable or more from the normal operating conditions, the property of a fault is that it can't be controlled adequately by controllers.

2. Failure: Failure is a fault, it appears when the behaviour of a given system that is working under certain conditions is permanently interrupted.
3. Malfunction: Malfunction also is a fault. Unlike failure, malfunction is an intermittent irregularity in the behaviour of a given system.

Faults can be classified based on different criteria, they can be categorized based on the place of occurrence:

- Sensor faults: these are faults that occur at sensors.
- Actuator faults: Here actuators are the faulty components.
- Plant faults: these are faults that occur in the plant components.

Faults can also be categorized based on the form of the fault itself:

- Abrupt faults: these faults behave as a step function.
- Incipient faults: these faults behave drift-like.
- Intermittent faults: these faults come with interrupts.

Figure 2 illustrates various types of faults based on their distinct form, referring to the temporal pattern or signature they exhibit over time. These include common fault patterns such as Abrupt, Incipient, and Intermittent faults. In each of these graphs, the y-axis represents the magnitude of the fault, while the x-axis denotes the progression of time.

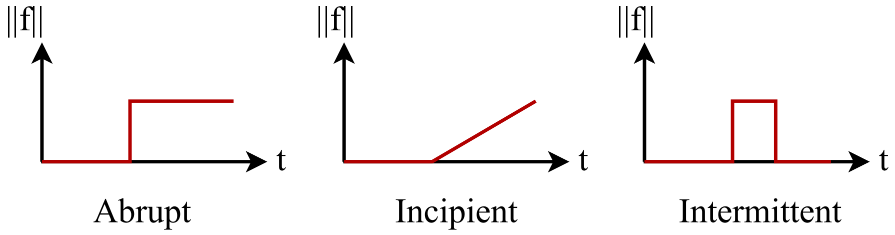

Figure 2: Different Types of Faults based on the Magnitude Deviation of the Faulty Variable.

Faults can be classified according to how the fault is added.

- Additive faults: faults change a given variable by addition; generally these are sensors' offsets.
$$Y(t) = U(t) + f(t) \tag{1}$$
- Multiplicative faults: these faults appear as a multiplication value; generally, these are changes in the process parameters.
$$Y(t) = f(t)\, U(t) \tag{2}$$

$f(t)$ is the fault function, $Y(t)$ is the faulty value, and $U(t)$ is the healthy value of the same system.

1.3 Process Monitoring

Process monitoring is a procedure associated with the following tasks: i) Fault detection. ii) Fault diagnosis. iii) Identification of faults. iv) Process recovery [3].

1. Fault Detection: The task determines whether the monitored system is faulty or not.

2. Fault diagnosis: This task determines the types of fault, the location and when the faults occurred, and the amplitudes of these faults.

3. Fault identification: This task is responsible for identifying the variables that contribute to the occurrence of the fault.

4. Process recovery: This is the final task in process monitoring, also known as intervention. It is responsible for removing the effect of the occurred faults.

1.4 Desirable Attributes of an FDD System

The FDD systems have some desired characteristics, these characteristics help to compare different FDD systems and to choose which one is suitable to use depending on the desired attributes which are listed as the following in [4]:

1. Quick detection and diagnosis: FDD systems should respond quickly to faults; when this system is too responsive and sensitive, it considers high-frequency influences and noises as faults, leading to higher FAR values. So, it is better not to be too greedy and increase the sensitivity of the FDD system to the fullest.

2. Isolability: It is the ability to differentiate between faults; if someone increases isolability too high the FDD system will have a deficient ability to reject model uncertainties and vice versa.

3. Robustness: The FDD system can reject the effects of noise and uncertainty of the model. One should balance between the isolability and robustness of a given FDD system.

4. Novelty Identifiability: It is one of the most important features of the FDD system. It is the ability to decide whether the monitored system is operating in healthy conditions or not and, if not, whether the faults are known or not.

5. Classification error estimate: This feature is responsible for evaluating the reliability of the monitoring system, which gives confidence in the diagnostic decisions.

6. Adaptability: The FDD system can adapt to changes in the monitored system because these changes are applied either by the user or due to changes in environmental conditions. So, the FDD system should adapt when more information about the system is available.

7. Explanation facility: FDD systems should provide explanations about faults as the cause and consequence relationship; in other words, they give the trail of the faults from the causes to their detection, and then FDD systems give why such hypothesis is made.

8. Modeling requirements: The modeling requirement of FDD systems should be minimized as much as possible.

9. Storage and computational requirements: FDD systems require algorithms and computations that take time for execution and memory space to save the needed information.

10. Multiple fault identifiability: The FDD systems can identify different multiple faults which is difficult because of the interactions between faults.

1.5 FDD Approaches

As was mentioned in the introduction, the FDD systems are categorized into two major methods which are Model-based approaches and Process history-based approaches, "Model-free approaches". Figure 3 presents different FDD approaches and shows their classification as well.

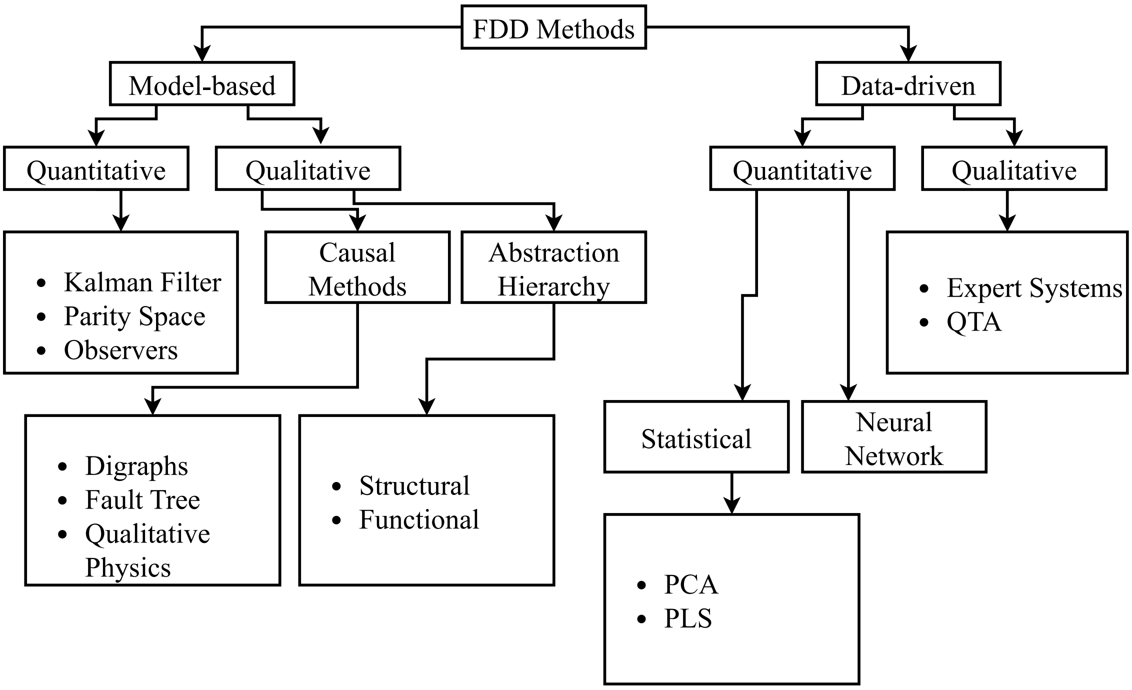

Figure 3: Classification of FDD Approaches [4].

1.5.1 Quantitative Model-based

Generally, quantitative model-based approaches are based on input/output or state-space models. These approaches consist of two steps; the first step is set to bring on the residuals between the actual and the expected behavior of the monitored system, whereas the second one is responsible for making the decision rule for diagnosis. Redundancy is used to check for residuals. Residuals are simply the discrepancies between the observed behavior of a system and its mathematical model prediction. They're typically zero when everything is working right, but they spike with significant values when a fault is present, making it easier to spot and pinpoint the problem. There exist two types of redundancy, analytical and hardware redundancies [4]. Hardware redundancy is not flexible because of its cost and the space required for implementation. Analytical redundancy is based on algebraic or temporal relationships between variables. It can be direct or temporal; direct redundancy is the result of finding algebraic relationships among variables of the system. The difference and differential relationships between the sensors' output and actuators' inputs are used to obtain the temporal redundancy.

The most used quantitative model-based approaches are:

- Observers: Observer-based FDD systems commonly utilize a bank of observers, each meticulously designed to exhibit selective sensitivity, allowing it to detect only certain faults. Since these observers rely on the generation of residuals derived from system redundancies, it is imperative that these residuals are inherently robust to unknown inputs and structured uncertainties [5].

- Parity Space: Parity Space is a powerful model-based technique for FDD that transforms system measurements into fault-sensitive residual signals, enabling the detection and isolation of abnormalities by observing deviations from expected behavior [6].

- Estimated Kalman Filter: Kalman filter (KF) is a recursive state estimation algorithm which is widely used in chemical and industrial processes. KF is used to estimate both sensors' and actuators' biases then they are compared to a threshold to decide whether there are faults or not [7].

Quantitative model-based approaches have some control over the residuals' behaviour. Unfortunately, the use of these approaches is highly related to the system's complexity, high dimensionality, process nonlinearity, and lack of good data about the process [7].

1.5.2 Qualitative Model-based

Qualitative model-based approaches characterize the relationships between the inputs and outputs of a system as qualitative functions, often centered around the behavior and interactions of different process units or components. This allows for modeling at a more abstract level, which is particularly useful when precise quantitative knowledge is unavailable [8]. Topographic search and symptomatic search are used for fault diagnosis. Figure 4 shows how these search methods are classified. Fault diagnosis often employs distinct strategies for pinpointing abnormalities. A topographic search analyzes deviations from a template of healthy operating conditions to identify mismatches and their locations within the system. Conversely, a symptomatic search directly correlates observed system symptoms (representing an abnormal state) with a library of known fault conditions to locate the fault [9].

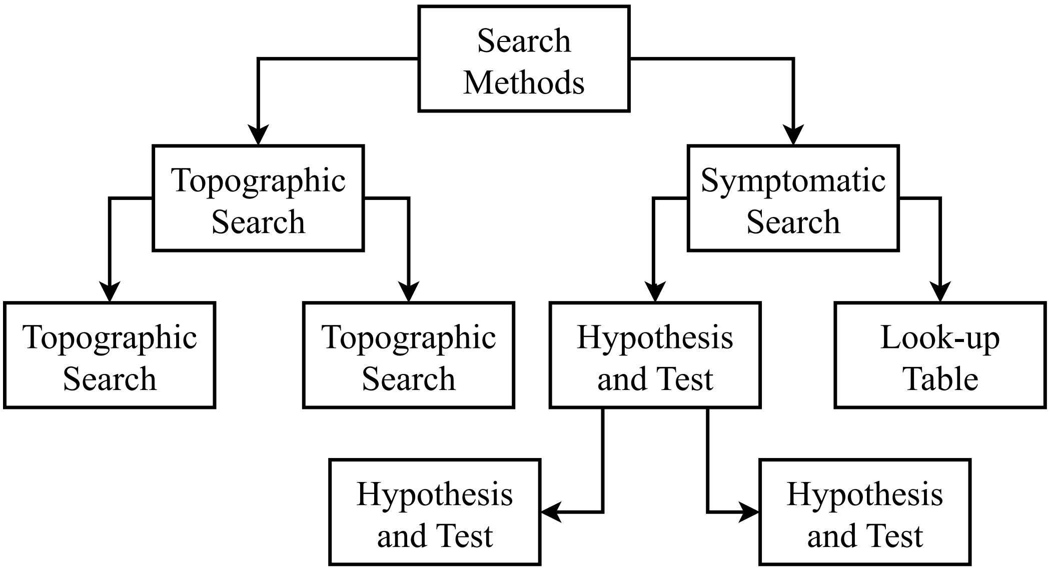

Figure 4: Different Search Methods [9].

As given in [9], the different qualitative approaches are:

- Digraphs Causal method: Digraphs are utilized to represent the cause-effect relationships within a given system, thereby modeling the process's intrinsic structure. These relationships are typically depicted as signed digraphs (SDGs), where nodes represent system variables and signed edges illustrate the qualitative (e.g., positive or negative) relationships between them. The core reasoning in this method involves identifying functional changes and their propagation patterns from faulty operating conditions. This allows digraphs to offer a powerful, intuitive, and explainable approach to FDD, proving particularly strong in fault isolation and understanding propagation paths, and thus bridging the gap between qualitative process knowledge and structured diagnostic analysis.

- Fault Tree Causal Method: The system's reliability and safety are surveyed by fault trees, these trees are logic trees that propagate from low-level events (i.e. primary events or faults) to the top-level events. They consist of layers of nodes with each node having one of the logic operators, unlike SDGs which only use **OR** operator. Fault trees are built following these steps:

  1. System definition.
  2. Fault tree construction.
  3. Qualitative and Quantitative evaluations.

- Qualitative Physics Causal method: Qualitative physics is broadly categorized into two main approaches. The first focuses on deriving qualitative differential equations, often referred to as confluence equations. The second approach aims to derive qualitative behavior directly from ordinary differential equations, which then serve as a valuable knowledge source.

- Structural Abstraction Hierarchy: Structural hierarchies illustrate how the information is connected between the system and its subsystems.

- Functional Abstraction Hierarchy: Unlike structural hierarchies, functional hierarchies characterize the means-end relationships between a system and its subsystems.

Qualitative model-based FDD approaches are particularly advantageous when dealing with uncertainty, noise, and the need for human-understandable diagnoses, making them valuable complements or alternatives to purely quantitative methods, especially in complex or ill-defined systems [10]. While qualitative model-based FDD excels in interpretability and robustness to uncertainty, its main limitations stem from the inherent loss of numerical detail, which can lead to ambiguity, limited resolution, and challenges in precisely detecting subtle faults or handling complex dynamics [8].

1.5.3 Drawbacks of Model-based FDD Approaches

The most common drawback of model-based approaches is the model complexity, complex processes can be very challenging to model accurately let alone the sensitivity to uncertainties in the parameters disturbances from outside the process. These approaches are designed for certain conditions, need to be updated regularly, and are more expensive than model-free approaches. Model-based approaches may not be robust enough for new faults that were not taken into account during the modelling of the monitoring system [11].

1.5.4 Model-free

Unlike the model-based approaches, the priori knowledge of a given system is obtained from a large amount of process data that has been transformed using feature extraction which may be qualitative or quantitative [12]. The model-free approaches can be classified as:

- Expert systems: Expert systems represent a category of qualitative, model-free approaches specifically designed for fault detection and diagnosis. These systems are highly dedicated and specialized to solve particular types of problems. An expert system's architecture typically comprises several key components: knowledge acquisition, a chosen knowledge representation scheme, the coding of knowledge in a knowledge base, and the development of input-output interfaces alongside inference procedures for diagnostic reasoning. A significant advantage of expert systems in FDD is their ease of development, coupled with their transparent reasoning capabilities. Furthermore, they can effectively reason under uncertainties and provide explicit explanations for their diagnostic conclusions [5].

- Qualitative Trend Analysis (QTA): QTA is a qualitative, model-free approach highly valuable in FDD and supervisory control. It serves as a potent tool for explaining significant process events, performing fault diagnosis, and predicting future system states. However, a notable limitation of QTA is its reliance on filters, which can unfortunately distort the underlying qualitative information of the process [13].

- PCA and Partial Least Squares (PLS): PCA and PLS are robust statistical methods utilized for quantitative feature extraction from process history data matrices in FDD. These matrices typically encompass all relevant process variables. PCA operates by orthogonally decomposing the covariance matrix of the process data. Its primary objective is to reduce the dimensionality of the data matrix while preserving the majority of the data's variability and essential process characteristics.In contrast, PLS employs two distinct data matrices from the process: one containing process variables and another holding related product quality variables. PLS aims to simultaneously model and compress the relationships between these two sets of variables. It achieves this by extracting latent variables that not only explain significant variability within the process variables matrix but also capture the variation most predictive of the product quality variables in the second matrix. This unique ability makes PLS particularly powerful for process monitoring where quality is a key concern, distinguishing it from PCA's unsupervised focus on maximizing variance alone [14].

- Statistical Classifiers: Statistical classifiers constitute another category of quantitative, model-free approaches widely employed in FDD. Their utility stems from the inherent nature of FDD as a classification problem, where the goal is to categorize system states as normal or various types of faulty conditions. This approach typically leverages a range of classification algorithms, commonly incorporating principles such as Gaussian density functions for probabilistic modeling and Euclidean distances as a measure of similarity or dissimilarity between data points [15].

- Neural Networks (NN): NN constitute a class of non-statistical, quantitative approaches widely employed in model-free FDD. Their application in FDD can be broadly categorized by two main dimensions: their network architecture (e.g., sigmoidal, radial basis function networks) and their learning strategy (e.g., supervised or unsupervised learning). NN offer the advantage of constructing 'black box' models directly from process data, thereby circumventing the need for explicit first-principles system models, which can be particularly beneficial for complex systems where detailed physical models are difficult to obtain [16]. However, a significant limitation is their substantial data requirement for achieving robust performance, as their effectiveness is largely confined to the range of the training dataset [16].

Table 1 shows the difference between different approaches in terms of desirable attributes [12].

Table 1: Comparison Between Different FDD Approaches

Attributes	Observers	Digraphs	Expert Systems	QTA	PCA	Neural Network
Quick Detection and Diagnosis	suitable	not assessed	suitable	suitable	suitable	suitable
Isolability	suitable	not suitable	suitable	suitable	suitable	suitable
Robustness	suitable	suitable	not suitable	suitable	suitable	suitable
Novelty Identifiablity	not assessed	suitable	not suitable	not assessed	suitable	suitable
Classification Error	not suitable	not suitable	not suitable	not suitable	not suitable	not suitable
Adaptability	not suitable	suitable	suitable	not assessed	not suitable	not suitable
Explanation Facility	not suitable	suitable	suitable	suitable	not suitable	not suitable
Modelling Requirement	not assessed	suitable	suitable	suitable	suitable	suitable
Storage and Computation	suitable	not assessed	suitable	suitable	suitable	suitable
Multiple Faults Identifiability	suitable	suitable	not suitable	not suitable	not suitable	not suitable

As it is seen from table 1, the choice of the method used for the FDD system is based on the requirements needed in monitoring the system.

1.5.5 Drawbacks of Model-free FDD Approaches

Model-free approaches depend entirely on the quality and quantity of the data set used for the training and validation part, so they lack physical insights and act like a black box system. Since these approaches are implemented in a hardware system they need to be fast enough to process the current samples and wait for the new ones. Feature extraction tasks are crucial in building the model so they must be extracted effectively. If some new type of fault is present then monitoring systems should be retrained [3].

1.6 Statistical Process Monitoring

The repetition of taking the measurements multiple times for the same process under the same conditions creates variations, common cause variation is a term used in Statistical Process Monitoring (SPM) for the variation obtained or expected to occur based on statistical distribution, special cause variation on the other hand occurs due to change in the process variables and they determine as unnatural variations such as faults [17]. SPM is a tool used to decide which of these variations are presented in the monitored system and to accomplish this task two limits are computed the upper control limit (UCL) of the distribution and its lower control limit (LCL) normal variations are presented within those limits and special cause variations are outside these limitations and the term upper is used to express that the limit is above the mean of the distribution and lower determines the limit under the mean of the distribution [17]. Multivariate Statistical Process Monitoring (MSPM) can monitor multiple variables simultaneously which is essential when dealing with processes where variables are interrelated and traditional univariate monitoring methods may fall short. MSPM offers an accurate and comprehensive monitoring approach and it can detect deviations in the process that might not be apparent when variables are treated individually. The most common techniques related to MSPM are the T^2 index and the Q index which are largely used in model-free process monitoring [14].

1.6.1 Monitoring Indices

Monitoring indices are crucial to understanding the health of a given system, providing information on various anomalies that could be present. In MSPM, these statistical indices are used to track the complete behavior of complex systems. Among these, the most fundamental monitoring indices include:

- Hotelling's T^2 index: This index evaluates the variation in the principal component subspace.
- Prediction Square Error (SPE) or Q index: This index is responsible for evaluating the variation in the residuals subspace.
- Combined index φ index: this index evaluates the variation in both principal components subspace and residuals subspace at the same time.

Usually, one of these indices is used for process monitoring, but for some application more than one index can be used.

1.6.2 FDD Performance Metrics

For FDD evaluation of system performance, different metrics were introduced, these metrics are used on a given MSPM distribution with its UCL, the metrics are related and changes in some of them affect the others [[18]]. False Alarm Rate (FAR) which gives a percentage of how much healthy samples act as faulty as in (3). Missed Detection Rate (MDR) used to see how much faulty samples act as healthy ones as in (4). Detection Time Delay (DTD) which is the time (or number of samples) required for the monitoring system to detect the fault as in (5).

$$FAR\,(\%) = \frac{NF}{NOC} * 100 \tag{3}$$

$$MDR\,(\%) = \frac{FN}{FOC} * 100 \tag{4}$$

$$DTD = t_d - t_o \tag{5}$$

where NF is the non-faulty samples acting as faulty ones, FN is the faulty samples acting as non-faulty ones, NOC is the total samples of normal operating condition, FOC is the total faulty samples, t_d is the detection time where t_o is the occurrence time.

1.6.3 Cost Functions

The cost functions proposed in this study are designed to optimize the monitoring model by focusing on reducing the number of abnormal samples. These functions play a critical role in parameter selection for the monitoring model and are also used to evaluate and compare the performance of different monitoring approaches.

For monitoring systems based on a single monitoring index (T^2, Q, or φ), the cost function utilized is defined by (6). This cost function serves multiple purposes:

- Parameter Optimization: It helps in determining the optimal parameters for the monitoring model. By minimizing the cost function, one can select parameters that reduce the inclusion of abnormal samples and enhance the model's effectiveness.
- Performance comparison: The cost function allows for the comparison of various monitoring models. By evaluating the values of the cost function for different models, it is possible to assess which model performs better in terms of handling and reducing abnormal samples.

$$J_s = a_1 FAR_s\% + a_2 MDR_s\% + a_3 \left(1 - e^{-0.1DTD_s}\right) 100\% \tag{6}$$

The weighting factors a_1, a_2, and a_3 are introduced to adjust the emphasis placed on different metrics within the cost function. When these factors are set equally, it means that all metrics are considered equally important. DTD is an integer because it represents the number of samples from the occurrence of the fault until detection, to ensure that DTD is comparable to the percentage metrics ($FAR_s\%$ and $MDR_s\%$), it is transformed using the exponential function $1 - e^{-0.1DTD_s}$. This transformation normalizes DTD so that it falls within a range similar to percentage metrics, the value of 0.1 was used such that the normalized values has an almost similar changes like the other two metrics, and this value is selected empirically. This normalization helps in preventing any single metric from disproportionately influencing the overall cost function value.

For a more comprehensive evaluation of monitoring systems, a general cost function is proposed. This cost function, denoted by J, represents the mean performance across all monitoring indices, providing a global view of the algorithm's effectiveness. The purpose of this general cost function is to enable users to assess the overall performance of a given algorithm based on multiple indices simultaneously.

In this approach, each monitoring index contributes equally to the overall cost function. By averaging the performance metrics across all indices, this cost function provides a balanced measure of the algorithm's performance, accounting for the various aspects of fault detection and monitoring.

The general cost function (7) allows users to compare different algorithms based on their performance across all relevant indices, rather than focusing on any single index. This comprehensive assessment helps in identifying algorithms that offer a well-rounded performance and are effective across multiple aspects of monitoring.

$$J = \frac{1}{3}\left(J_{T^2} + J_Q + J_\varphi\right) \tag{7}$$

1.7 Conclusion

This chapter provides a foundational overview of FDD, outlining various approaches and their underlying principles. It discusses the different methods available for FDD and emphasizes the importance of selecting the appropriate approach based on the specific process being monitored. Model-based approaches are often more suitable for certain processes due to their ability to leverage process models for fault detection, whereas model-free approaches may be more appropriate for others, depending on their characteristics and requirements. The chapter also highlights the significance of model-free methods, such as Model-Free Statistical Process Monitoring (MSPM), which plays a crucial role in fault detection without relying on a process model.

To conclude the chapter, a cost function is introduced, designed to assess the performance of monitoring systems by focusing on the number of abnormal samples detected. This cost function serves as a tool to evaluate the effectiveness of different monitoring systems, providing a quantitative measure to compare their performance in detecting faults.

2 Multivariate Statistical Approaches

2.1 Introduction

PCA and Kernel PCA (KPCA) are unsupervised process monitoring methods based on process history data, they are flexible and easy to use without the need for optimization or a large data set like NN. This chapter details PCA and KPCA as fault detection tools, the drawbacks of each one of them, and how to use different indices and metrics to monitor a given system.

2.2 PCA and Its Nonlinear Extensions: Challenges and Solutions

MSPM is used to monitor and analyze a given system that has multiple interrelated variables; by doing this, it can maintain the quality and stability of the process. MSPM is widely used in different fields. PCA is one of the widely used methods in MSPM, it gained this reputation because of its efficiency, simplicity, and flexibility [19]. MSPM uses PCA's Principal Components (PC) to identify and detect abnormal behavior in the given monitored system.

The idea of PCA is to reduce the dimensionality of the process data while preserving most of the variability of this data [20]. This reduction is conducted using orthogonal decomposition of the covariance matrix and selecting some PCs from the original data set. The PCA technique is performed while assuming that the monitored system's data have linear characteristics between its variables, which is not the case for large and complicated industrial systems. Many alternative PCA techniques have been introduced to overcome this kind of problem. Proposed by *Kramer et al.* [21], Nonlinear PCA excels at modeling complex or curved data relationships, offering greater accuracy and potentially fewer components than traditional PCA. However, its reliance on sophisticated mathematical functions (e.g. artificial NN) means that it demands sufficient data to avoid overfitting and ensure reliable estimations. In the same context as in the previous paper, *Tan & Mavrovouniotis* [22] uses NN with a single hidden layer network unlike *Kramer et al.* [21] which uses NN of three hidden layers. *Dong and McAvoy* [23] introduced a Nonlinear PCA method that merges principal curves with NN modeling to capture nonlinear data relationships, a crucial advantage in complex real-world applications. However, this approach can be computationally intensive and its performance is highly dependent on the proper selection of network architecture and parameters. *Hiden et al.* [24] proposed an alternative PCA technique that uses Genetic Algorithm, making it suitable for various complex systems. However, a limitation is the necessity to predetermine the number of nonlinear components, which reduces its flexibility compared to standard PCA and may require more robust mathematical functions for challenging data. *Scholkopf* [25] introduced the kernel PCA (KPCA), its idea is simple, which starts by mapping data into a higher-dimensional Hilbert space, called feature space, and then applies conventional PCA on this mapped data. The main advantage of KPCA over other solutions is that it does not require optimization since it is based on the decomposition of eigenvalues as conventional PCA [25]. KPCA can easily deal with the nonlinear characteristics of a given data set. Unfortunately, KPCA's kernel matrix size is related to the number of samples in the training data set so if the data under study have a large number of observations then the KPCA algorithm becomes disadvantageous because the large matrix can affect the accuracy of eigenvalue computation as stated in [25], furthermore, the more observations the data set has, the larger the execution time and memory storage space become. The time complexity of the KPCA algorithm is given as $\mathcal{O}(n^3)$ and the space complexity is $\mathcal{O}(n^2)$ where n is the number of samples in the training data set. To overcome this kind of problem, the reduced KPCA (RKPCA) is introduced. RKPCA is composed of two parts; the first part is responsible for data reduction, which means it reduces the number of observations using a certain technique, while the second part builds the KPCA model upon the reduced data obtained from the first part.

2.3 PCA Monitoring Technique

2.3.1 Definition and Mathematical Formulation

PCA is an MSPM technique that aims to reduce the dimensionality of the process data set by mapping this data into lower dimensional space where only uncorrelated variables remain possessing most of the process information, this is done by an orthogonal decomposition so that a set of PC represents the same system [26].

Let $X_o \in R^{n \times m}$ be the original data set collected from m measurements variables for n times (n is the number of observations). Before carrying on with the PCA procedure X_o must be transformed, this data is normalized to zero mean and unit standard deviation. This step is important because variables with larger magnitudes or variances can disproportionately influence PC, causing them to overshadow other potentially more important features. Normalization ensures that all variables contribute equally, allowing PCA to accurately identify the most significant underlying relationships across the data set. Equation (8) is the normalization formula used in this thesis.

$$\tilde{X} = \frac{X_o - V_m}{V_s} \tag{8}$$

Where $\tilde{X}$ is the normalized data set used in PCA approach, V_m is vector of the means of m variables, and V_s is the standard deviation.

The following step is the computation of the covariance matrix, C, because PCA is based on this matrix decomposition as in equation (9).

$$C = \frac{1}{n-1}\tilde{X}^T\tilde{X} \tag{9}$$

The singular value decomposition is used to rewrite the covariance matrix using eigenvalues and eigenvectors of C as in (10).

$$C = P\Lambda P^T \tag{10}$$

Eigenvectors of C are the column vectors of P, and the diagonal elements of Λ are eigenvalues in decreasing order.

The scores matrix, T, is the mapped matrix of the input data set using loading vectors, P, to the new dimension space [18, 27].

$$T = \tilde{X}P \tag{11}$$

Using PCA, only some eigenvalues and their corresponding eigenvectors are selected to characterise most of the information from the input data set. This number, l, is known as the PCs of the system, the first PC contains most variance of the data then followed by the second PC and so on. The eigenvalues, eigenvectors, and scores matrix of the training data set is then given by (12), (13), and (14).

$$\Lambda = \begin{bmatrix} \hat{\Lambda}_{l\times l} & 0_{l\times(m-l)} \\ 0_{(m-l)\times(l)} & \bar{\Lambda}_{(m-l)\times(m-l)} \end{bmatrix} \tag{12}$$

$$P = \begin{bmatrix} \hat{P}_{m\times l} & \bar{P}_{m\times(m-l)} \end{bmatrix} \tag{13}$$

$$T = \begin{bmatrix} \hat{T}_{m\times l} & \bar{T}_{m\times(m-l)} \end{bmatrix} \tag{14}$$

Normalised data can be divided into model variation, $\hat{X}$, obtained from PC loading vectors and non-model variation, $\bar{X}$, obtained from residual loading vectors as explained in (15) and (16).

$$\tilde{X} = \tilde{X}\hat{P}\hat{P}^T + \tilde{X}\bar{P}\bar{P}^T = \tilde{X}\hat{P}\hat{P}^T + \tilde{X}\left(I - \hat{P}\hat{P}^T\right) \tag{15}$$

$$\tilde{X}_{(n\times m)} = \hat{X}_{(n\times l)} + \bar{X}_{(n\times(m-l))} \tag{16}$$

PCA maps data to the PC subspace that contains most of the characteristics and variations of the original data set, and the remaining data is assigned to the residual subspace [28].

2.3.2 PCs Selection Techniques

Although the covariance matrix in PCA generates m components, a carefully chosen subset of l PCs ($l < m$) is utilized to develop a fault detection system that is more focused, efficient, and sensitive. This selection allows to concentrate on meaningful variations while filtering out noise. The choice of l PC directly determines the dimensionality of the PC subspace and how much of the input data's characteristics are retained for a given application, with various techniques available to make this determination.

- Cumulative Percent Variance (CPV): The number l is the smallest value for which $CPV\,(l)$ is greater than a defined limit CPV_{limit} [19].

$$CPV\,(l) = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{j=1}^{m} \lambda_j} \times 100 \geqslant CPV_{limit} \tag{17}$$

λ_i's are eigenvalue from Λ. l is the smallest number which results $CPV\,(l) \geqslant CPV_{limit}$.

- Scree Plot: the scree plot uses a graph between eigenvalues and the number of PC, it explains the variance associated with each PC. To apply this method, start by ploting eigenvalues in descending order, then search for the elbow pointand the number of PC before this point is the number of selected number of PC l [29].
- Kaiser Criterion: This is a straight forward technique which retain eigenvalues that are greater than 1.00 and use them as PC [30].
- Cross Validation: It minimizes the prediction error in supervised learning context. It starts by splitting data to a training and testing data sets then performs PCA with different numbers of PC. After that, it evaluates the performance of the training set using testing set. The appropriate number of PC is the one with the best performance [31].

In this study, CPV technique is used due to its simplicity and it is the widely used one for fault detection techniques.

2.3.3 Monitoring Indices

The Hotelling's T^2-index is responsible for determining deviations of variables from their means, this is evaluated in the PC subspace [32]. The T^2-index is given by (18).

$$T^2 = \tilde{x}^T \hat{P} \hat{\Lambda}^{-1} \hat{P}^T \tilde{x} \tag{18}$$

For MSPM, a limit of this index is needed to decide whether the system's behaviour is normal or abnormal. The UCL of the T^2-index is given based on Fisher-Snedecor distribution, $\mathcal{F}_\alpha$, as shown in (19).

$$T^2_\alpha = \frac{(n^2-1)\, l}{n\,(n-l)} \mathcal{F}_\alpha\,(l, n-l) \tag{19}$$

l & $n-l$ are the distributions's degrees of freedom and α is the significance level.

The Q-index is responsible for monitoring residuals of the PCA model. Unlike the T^2-index, Q-index evaluates variations in Residuals subspace [32]. The Q-index and its UCL are given by (20) and (21), respectively.

$$Q = \tilde{x}^T \bar{P} \bar{P}^T \tilde{x} \tag{20}$$

$$Q_\alpha = \frac{\sigma_Q^2}{2\mu_Q} \chi^2 \left(\alpha, \frac{2{\mu_Q}^2}{\sigma_Q^2} \right) \tag{21}$$

where σ_Q^2 represents the variance of Q-index, and μ_Q is its mean. χ^2 is the Chi-squared distribution.

Another index was introduced by [33], this index is known as φ-index which is the weighted linear combination of the other two indices (T^2 and Q).

$$\varphi = \frac{T^2}{T_\alpha} + \frac{Q}{Q_\alpha} \tag{22}$$

the UCL of this index is given based on χ^2 distribution, g and h are given by the following equations.

$$\varphi_\alpha = g\chi^2\left(\alpha, h\right) \tag{23}$$

$$g = \frac{\frac{l}{(T_\alpha^2)^4} + \sum_{i=l+1}^{n} \frac{\lambda_i^2}{Q_\alpha^4}}{(n-1)\left(\frac{l}{(T_\alpha^2)^2} + \sum_{i=l+1}^{n} \frac{\lambda_i}{Q_\alpha^2}\right)}$$

$$h = \frac{\left(\frac{l}{(T_\alpha^2)^2} + \sum_{i=l+1}^{n} \frac{\lambda_i}{Q_\alpha^2}\right)^2}{\frac{l}{(T_\alpha^2)^4} + \sum_{i=l+1}^{n} \frac{\lambda_i^2}{Q_\alpha^4}}$$

2.3.4 Fault Detection Using PCA

This subsection explains the application of the PCA approach for FDD. This approach is fundamentally divided into two parts: an offline (training) part in which a PCA model of the monitored process is constructed and an online (monitoring) part in which new incoming data are examined using the monitoring model.

The algorithm 1 shows the inputs and outputs of the PCA algorithm for the offline part. From the same algorithm, it is seen that PCA starts by normalizing data with zero mean and unit variance to ensure that the used data has the same range because variables with large range and means can affect the model built for the monitoring, then computes the covariance matrix and deduces the sorted eigenvalues and eigenvectors of the system. Next, the CPV method is used for the selection of PCs, the ultimate goal is to achieve strong monitoring performance, characterized by an FAR (e.g., 5%), a minimal MDR (e.g., less than 5%), and an acceptable DTD. The most important guideline here is to avoid "greed", trying to include too many PCs risks overfitting the model to the training data. This means the model might inadvertently capture noise and minor fluctuations specific to the training set, leading to an overly sensitive model that triggers frequent false alarms when applied to new, unseen data. Near the end, PCA computes different monitoring indices and their limits. In the end, the monitoring model is stored; this model contains the necessary parameters and data used in the online part of the PCA approach.

Algorithm 1 Offline part of PCA for Fault Detection

```
Input: Data matrix $X_o \in R^{n \times m}$.
Output: Monitoring model.
Step 1: Standardize data
Normalize and center $X_o$ as in (8)
Step 2: Compute the covariance matrix
Compute $C$ as in (11), $C \in R^{m \times m}$
Step 3: Compute eigenvalues $\Lambda$ and eigenvectors $P$
$[P, \Lambda] = \text{eig}(C)$
Sort eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ and reorder eigenvectors accordingly.
Step 4: Select PC
Choose $l$ largest eigenvalues $\hat{\Lambda}$ and corresponding eigenvectors $\hat{P}$ using CPV as in (17).
Step 5: Compute different monitoring indices
Compute $T^2$, $Q$, and $\varphi$ indices using (18), (20), and (22).
Step 6: Compute the UCL of these indices
Compute ${T_\alpha}^2$, $Q_\alpha$, and $\varphi_\alpha$ limits using (19), (21), and (23).
Step 7: Store monitoring model
```

Algorithm 2 presents the online part of the PCA approach, it starts by scaling the newly acquired samples with the mean and standard deviation of the training data set and then projects these newly collected and scaled data onto the PC subspace. After this projection, it calculates the different monitoring indices and then compares them to their limits from the offline part. In the end a decision about the system is taken upon these comparisons, if at least one of these indices exceeds its limit then a fault is declared otherwise the system is healthy and no alarm is triggered. One should take note that it is not always necessary to use all the monitoring indices, some applications may require just one of them.

Algorithm 2 Online part of PCA for Fault Detection

```
Input: New observations $X_\tau \in R^{\tau \times m}$, monitoring model
Output: Fault detection result
Step 1: Standardize $X_t$
$\tilde{X}_\tau = \frac{X_\tau - V_m}{V_s}$                ▷ Standardize using training data mean and std
Step 2: Project onto PC
$T_\tau = \tilde{X}_\tau \hat{P}$                          ▷ Projection onto the PC subspace
Step 3: Compute different indices
Compute different indices using new observations and equations (18), (20), and (22).
Step 5: Fault Detection
if $T^2 > T^2_\alpha$ or $Q > Q_\alpha$ or $\varphi > \varphi_\alpha$ then
    Fault Detected
else
    No Fault Detected
end if
```

A flowchart of the PCA algorithm, presented in figure 5, offers a generalized representation of its operation in fault detection. In particular, this flowchart is adaptable for KPCA and RKPCA methods with appropriate modifications. The blue section of this flowchart represents the offline part of the PCA algorithm (Algorithm 1), while the green section depicts the online part (Algorithm 2). The online part operates in an iterative manner by processing new data immediately upon arrival.

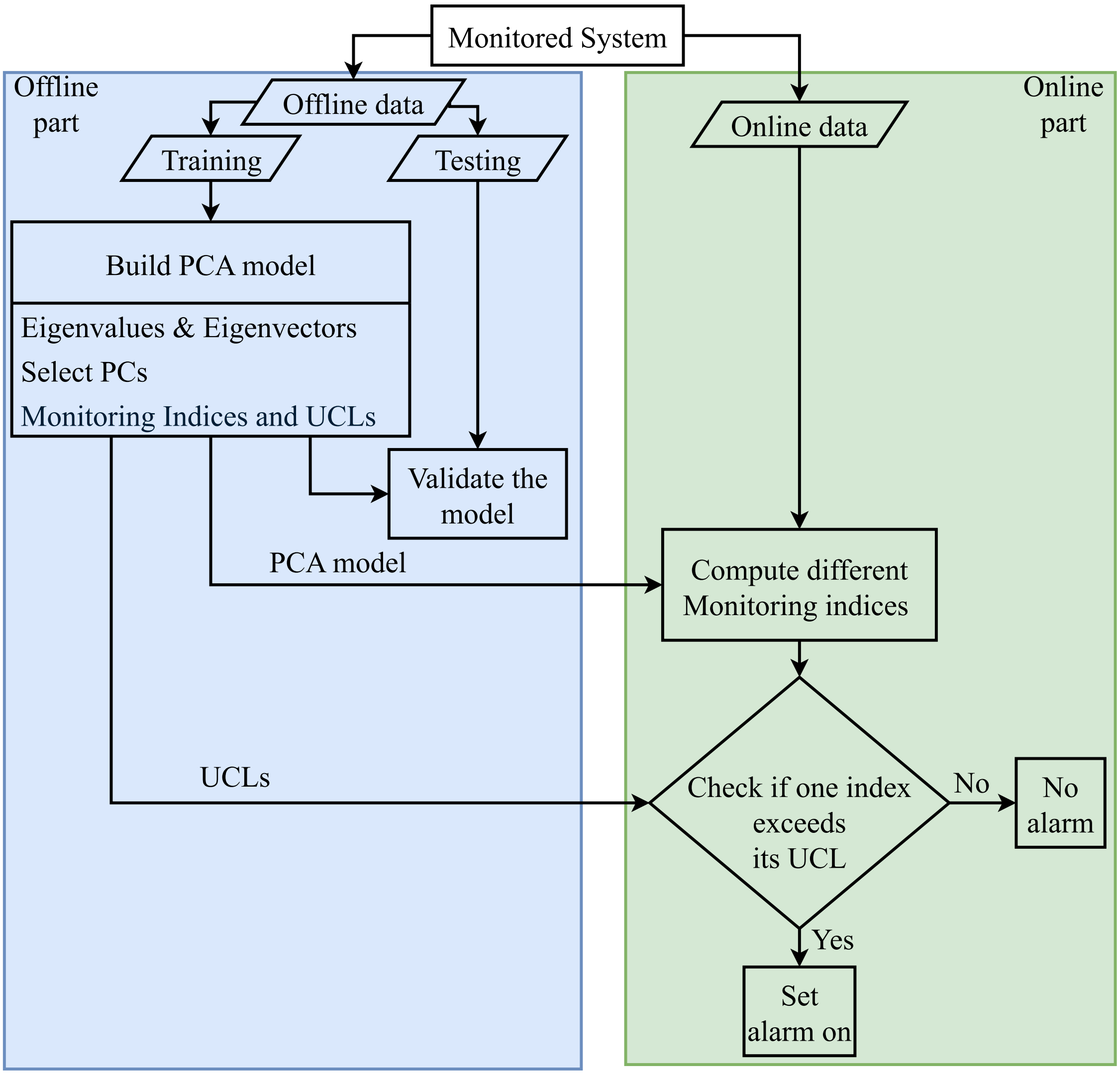

Figure 5: Flowchart of FDD based on PCA.

2.3.5 Disadvantages of PCA

Although PCA is widely used and has a good reputation in FDD it requires some assumptions on the data collected from the monitored system, PCA requires that this data has linear characteristics between its variables, and the mapping performed by the PCA is a linear mapping which results in a loss of nonlinear information from the training data set during the mapping [34], also the small number of PCs affects the ability to detect nonlinearities within the data set [25]. Another main drawback is that PCA is designed for time-invariant processes because it can not adapt the monitoring model to the online part.

2.4 KPCA Monitoring Technique

2.4.1 Definition and Mathematical Formulation

KPCA is a powerful variation of PCA designed to handle nonlinearities in data set. This is achieved by first transforming the data into a higher-dimensional feature space. Nonlinear relationships in a lower-dimensional space can become linear relationships in a sufficiently higher-dimensional space [35]. Once data is transformed into this new feature space, those formerly tangled nonlinear patterns are "unfolded" or "untangled" in such a way that they become linearly separable. This means separating different groups of data points in this higher dimension can be achieved by a straight line or a flat plane (a hyperplane).

Let $\tilde{X} \in R^{n\times m}$ be the normalized data set obtained from the monitored process using equation (8), let's define a mapping function Φ as: $\Phi : R^m \longrightarrow F, \;\; \tilde{X} \longrightarrow \tilde{X}_F$. F is a Hilbert space and its dimensionality is larger than the one of the input space and it could be infinite [25].

To compute the covariance matrix, it is assumed that the mapped data set using Φ is centred, $\sum_{k=1}^{n} \Phi\left(\tilde{x}_k\right) = 0$ and $\tilde{x}_k$ is a row vector from $\tilde{X}$ [25]. The covariance matrix in F is then given by (24).

$$C_F = \frac{1}{n}\sum_{j=1}^{n} \Phi\left(\tilde{x}_j\right).\Phi\left(\tilde{x}_j\right)^T \tag{24}$$

As conventional PCA, KPCA is also based on eigenvalues decomposition as shown in equation (25) [25].

$$P_F\Lambda_F = C_F P_F, \;\; \Lambda_F \geq 0 \;\; \& \;\; P_F \in F \tag{25}$$

All solutions P_F with $\Lambda_F \neq 0$ lie in span of $\left[\Phi\left(\tilde{x}_1\right), \; \Phi\left(\tilde{x}_2\right), \cdots, \; \Phi\left(\tilde{x}_n\right)\right]$, as a result the following equations are considered [25].

$$\Lambda_F\left(\Phi\left(\tilde{x}_k\right)P_F\right) = \left(\Phi\left(\tilde{x}_k\right)C_F P_F\right), \qquad k = 1\cdots n \tag{26}$$

$$P_F = \sum_{i=1}^{n} a_i \Phi\left(\tilde{x}_i\right), \qquad i = 1\cdots n \tag{27}$$

Equation (28) is obtained by combining the previous two equations.

$$\Lambda_F \sum_{i=1}^{n} a_i\left(\Phi\left(\tilde{x}_k\right)\Phi\left(\tilde{x}_i\right)\right) = \frac{1}{n}\sum_{i=1}^{n} a_i\left(\Phi\left(\tilde{x}_k\right)\sum_{j=1}^{n}\Phi\left(\tilde{x}_j\right)\right)\left(\Phi\left(\tilde{x}_j\right)\Phi\left(\tilde{x}_i\right)\right), \;\; k = 1\cdots n \tag{28}$$

Let's define matrix $K_{n\times n}$ such that $K_{n\times n} = \Phi\left(\tilde{x}_j\right)\Phi\left(\tilde{x}_i\right)$, $i\ \ \&\ \ j = 1\cdots n$. By substituting in equation (28), it can be written as:

$$n\ \Lambda_F\ K\ o = K^2\ o, \quad\ o = [a_1,\ a_2, \cdots,\ a_n] \tag{29}$$

Which can be simplified as:

$$n\ \Lambda_F\ o = K\ o \tag{30}$$

The last equation denotes an eigenvalue problem to be solved, Λdenotes non-zero positive eigenvalues of K and o correspond to eigenvectors and these eigenvectors are normalized $o_k.o_k = \frac{1}{\lambda_k},\ \ k = 1\cdots n$.

The mapping function Φ does not necessarily need to be known or used explicitly, the mapping is replaced or substituted by a kernel function to compute the matrix $K_{n\times n}$, this is known as kernel trick [25].

$$K_{n\times n} = \left[\kappa\left(\tilde{x}_i,\ \tilde{x}_j\right)\right] = \left[\Phi\left(\tilde{x}_i\right)\Phi\left(\tilde{x}_j\right)\right]\quad i, j = 1\cdots n \tag{31}$$

$\kappa\left(\tilde{x}_i,\ \tilde{x}_j\right)$ is a kernel function, a kernel function is used only if it satisfies Mercer's theorem and this is known as a kernel trick [25]. Mercer's theorem is given in Appendix A. There exist different kernel functions, these are the most popular ones:

1. Polynomial: $\kappa\left(\tilde{x}_i,\ \tilde{x}_j\right) = \left(\tilde{x}_i\ \tilde{x}_j\right)^d,\ \ d \geq 1\ \ \&\ \ d \in N$. This function satisfies Mercer's theorem for positive nonzero integer d.

2. Radial Basis Function (RBF): $\kappa\left(\tilde{x}_i,\ \tilde{x}_j\right) = exp\left(-\frac{\|\tilde{x}_i - \tilde{x}_j\|^2}{2\sigma^2}\right),\ \ 2\sigma^2 > 0$. This function satisfies Mercer's theorem for and nonzero positive real number $2\sigma^2$.

3. Sigmoid kernels: $\kappa\left(\tilde{x}_i,\ \tilde{x}_j\right) = tanh\left(f\left(\tilde{x}_i\ \tilde{x}_j\right) + p\right)$. this function satisfies Mercer's theorem for a few combinations of numbers $p\ \&\ f$.

RBF is used in this thesis because of its flexibility, it is widely used, and it has a wide range of allowed hyper-parameter $2\sigma^2$ [36]. The hyper-parameter is then given as $\sigma^2 = r\ m\ v^2$ where m is the dimensionality of input space, v^2 is the variance of input data, and r is empirically obtained for most unsupervised learning [37]. For the PC's selection, the CPV approach is used as presented in equation (17).

2.4.2 Monitoring Indices

Since KPCA is an alternative PCA approach and they are mostly similar, the same monitoring indices (T^2, Q, and φ indices) are used along with their UCLs [38], these indices are computed as in equations (32), (33), and (34).

$$T^2 = \kappa\left(\tilde{x}\right)^T\ \hat{P}_F\hat{\Lambda}_F^{-1}\hat{P}_F^T\kappa\left(\tilde{x}\right) \tag{32}$$

$$Q = \kappa\left(\tilde{x},\ \tilde{x}\right) - \kappa\left(\tilde{x}\right)^T\left(I_n - \hat{P}_F\hat{P_F^T}\right)\kappa\left(\tilde{x}\right) \tag{33}$$

$$\varphi = \frac{T^2}{T_\alpha^2} - \frac{Q}{Q_\alpha} \tag{34}$$

The corresponding UCLs are given as in equations (19), (21), and (23).

2.4.3 Fault Detection Using KPCA

The KPCA algorithm consists of two parts offline part where the KPCA model is built and an online part for fault detection.

Algorithm 3 presents the offline part of KPCA algorithm. KPCA starts by normalizing training data set with zero mean and unit variance then it computes the the kernel matrix using a selected kernel function. This kernel matrix needs to be centered before solving the eigenvalues problem, scale these eigenvalues and eigen vectors. After that, select the right number of PC and compute different indices and their limits. At the end, the necessary data used in the online part is stored.

Algorithm 3 Offline part of KPCA for Fault Detection

```
Input: Training data matrix $X_o \in R^{n \times m}$.
Output: Monitoring model.
Step 1: Standardize data
Normalize training data set with zero mean and unit variance using (8).
Step 2: Compute the Kernel Matrix
Choose a kernel function $\kappa(\tilde{x}_i, \tilde{x}_j)$ and compute the kernel matrix $K \in F^{n \times n}$ where $K_{ij} = \kappa(\tilde{x}_i, \tilde{x}_j)$
Step 3: Center the Kernel Matrix
$K_c = K - 1_n K - K 1_n + 1_n K 1_n$ where $1_n$ is a $n \times n$ matrix with all elements equal to $\frac{1}{n}$
Step 4: Compute Eigenvalues and Eigenvectors
$[P_F, \Lambda_F] = \text{eig}(\frac{1}{n} K_c)$
Sort eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ and reorder eigenvectors accordingly
Step 5: Select PC
Choose $l$ largest eigenvalues and corresponding eigenvectors using CPV.
Step 6: Compute monitoring indices
Compute different monitoring indices $T^2$, $Q$, and $\varphi$ using (32), (33), and (34).
Step 7: Compute limits of monitoring indices
Compute different monitoring indices limits ${T_\alpha}^2$, $Q_\alpha$, and $\varphi_\alpha$ using (19), (21), and (23).
Step 8: Store monitoring model
```

Algorithm 4 presents the online part of the KPCA algorithm. it normalizes the newly collected τ samples with the mean and standard deviation of the training data set and then computes the testing kernel matrix using both newly acquired data and the training data set after that it centres this testing kernel matrix. The different monitoring indices are then computed and compared to the limits obtained in the training part, if at least one of them exceeds its limit then a fault is declared.

Algorithm 4 Online part of KPCA for Fault Detection

```
Input: New observations $X_\tau \in R^{\tau \times m}$, monitoring model
Output: Fault detection result
Step 1: Standardize $X_\tau$  normalize using (8)
Step 2: Compute Kernel Matrix $K_\tau$
Compute the kernel matrix $\kappa(X_\tau, X) \in F^{\tau \times n}$. where $X$ is the scaled training data set
Step 3: Center $K_\tau$
$\tilde{K_\tau} = K_\tau - 1_\tau K - K_\tau 1_n + 1_\tau K 1_n$ where $1_\tau$ is a $\tau \times n$ matrix with all elements equal to $\frac{1}{\tau}$
Step 4: Compute different monitoring indices
compute $T^2$, $Q$, and $\varphi$ using (32), (33), and (34)
Step 5: Compute limits of monitoring indices
compute ${T_\alpha}^2$, $Q_\alpha$, and $\varphi_\alpha$ using (19), (21), and (23)
Step 6: Fault Detection
if $T^2 > {T_\alpha}^2$ or $Q > Q_\alpha$ or $\varphi > \varphi_\alpha$ then
    Fault Detected
else
    No Fault Detected
end if
```

2.4.4 Time and Space Complexities of KPCA

In this section, a detailed analysis of the time complexity of the KPCA algorithm is provided. The time complexity of the KPCA algorithm is primarily influenced by three major steps:

- Kernel, K, matrix computation
- Kernel matrix centring using

$$K_c = K - 1_n K - K 1_n + 1_n K 1_n$$

- Eigenvalues Decomposition

$$[P_F, \Lambda_F] = eig(\frac{1}{n} K_c)$$

The calculation of the kernel matrix for a given training data set with n samples has a time complexity of $\mathcal{O}(n^2)$ because the size of the kernel matrix is $n \times n$ whose elements are $\kappa(x_i, x_j),\ \ i, j = 1 \rightarrow n$. Since there are n^2 entries to compute, this step is $\mathcal{O}(n^2)$. The centering of the kernel matrix also requires $\mathcal{O}(n^2)$. This step involves matrix manipulations such as subtraction and addition, which are linear to the number of matrix elements. The eigenvalue decomposition for the kernel matrix is $\mathcal{O}(n^3)$ due to the nature of the used algorithms. The common QR iteration (used in MATLAB) primarily involves core linear algebra operations such as matrix multiplication, which inherently scale as $\mathcal{O}(n^3)$, which is the most computationally intensive part of the KPCA algorithm. Therefore, the overall time complexity of the KPCA algorithm is determined by the dominant term, which is the eigenvalue decomposition. Consequently, the overall time complexity of KPCA is $\mathcal{O}(n^3)$ [39].

The storage space complexity of the KPCA algorithm is influenced by several key steps in the algorithm. These steps are:

- Storing the kernel matrix K.
- Intermediate data storage.
- Eigenvalues and eigenvectors storage.

To store the kernel matrix, which is of size $n \times n$, the storage complexity is $\mathcal{O}(n^2)$. Similarly, storing the intermediate matrices, such as the centered kernel matrix, also requires $\mathcal{O}(n^2)$. For eigenvalue decomposition, the eigenvectors are stored in an $n \times n$ matrix, requiring $\mathcal{O}(n^2)$. The eigenvalues, although stored in a vector of size n, contribute a space complexity of $\mathcal{O}(n)$, which is negligible compared to $\mathcal{O}(n^2)$ for the eigenvectors. Therefore, the overall storage complexity of the KPCA algorithm is dominated by the space required for the kernel matrix and the eigenvectors. Consequently, the overall storage complexity of KPCA is $\mathcal{O}(n^2)$ [19].

2.4.5 Disadvantages of KPCA

KPCA is known to have a good monitoring performance and outperforms PCA in the case of nonlinear processes. This solution comes with a trade-off, to overcome nonlinearity in the process the monitoring system requires more data and with more data these drawbacks may be highlighted.

- High execution time because the time complexity of KPCA algorithm is $\mathcal{O}(n^3)$ because KPCA needs to solve an eigenvalues problem of an $n \times n$ matrix.
- More storage space required because the storage space complexity of KPCA algorithm is $\mathcal{O}(n^2)$, this complexity is the result of storing a $n \times n$ kernel matrix.
- For large data set the KPCA algorithm may face a problem solving appropriately the eigenvalues problem which can affect the monitoring performance [25].

all the three disadvantages stated above are related to the size of the training data set. RKPCA is an algorithm designed to overcome these specific disadvantages of KPCA without losing the ability to monitor nonlinear processes.

2.5 Conclusion

In this section, both KPCA and PCA were introduced and detailed to give a closer look at how both of them are designed for fault detection purposes. PCA was first introduced but it cannot perform well when data has nonlinear characteristics, KPCA then comes as a result to deal with this nonlinearity but unfortunately, it also creates new challenges when data has too many observations.

3 Proposed RKPCA Algorithms

3.1 Introduction

RKPCA reduces training data before applying the KPCA algorithm to avoid problems stated in the previous chapter, the reduction method is responsible for keeping most data information with fewer samples. The effectiveness of the reduction method will be seen in the monitoring performance. In this chapter, three proposed algorithms are introduced along some related works and a non-parametric homogeneity test is introduced toward the end of this chapter.

3.2 Related Work: Similar Approaches

Since the proposed algorithms are based on the RKPCA approach, some existing related works are presented to see what they utilize to reduce the size of the data set. One should take in mind that the reduction part must keep most information from the original data set to overcome the KPCA's limitations stated previously.

Euclidean distance RKPCA presented in [19] is based on similarity and Euclidean distance; here the Euclidean distance is used as a similarity measure, which means that the smaller the distance between two samples, the more similar they are and vice versa. This approach starts by selecting a threshold distance, and samples with a distance greater than this threshold are kept in the reduced data which is then used to build the monitoring model. After that, change the threshold and repeat to form another reduced data. The one with the best performance is used to build the final monitoring model. Euclidean distance RKPCA when used right can lead to decent monitoring performance compared to conventional KPCA. The limitation of this approach is that outliers or noise in one variable can greatly increase the Euclidean distance, making a slightly anomalous point appear very dissimilar to otherwise close clusters due to the amplification of large differences.

Reduced Rank RKPCA algorithm [40] is based on removing dependencies of variables in the feature space and retaining only a smaller number of observations. First, a row is selected from the training data set and the kernel vector is computed. Then select a second row and do the same as for the first one. Second, add the second kernel vector to the first one to form a matrix and check if this matrix is a full rank. Third, if this matrix is full rank, then keep this row in the kernel matrix; otherwise, delete it and move to the next vector in the training data set. Reduced Rank RKPCA offers significant advantages, including computational efficiency for online monitoring, a strong capability to handle nonlinear dynamics, and a wide applicability to various real-world systems, laying a crucial foundation for advanced condition monitoring. However, its primary disadvantages stem from the need to carefully define the number of observations retained, which leads to inflexibility once set.

K-means RKPCA Clusters algorithm [41] classifies the data set to a predefined number of disjoint clusters, and these clusters are represented by a cluster center for each one of them. It assigns the inputs to the nearest cluster center based on the mean squared error between the inputs and the cluster center. This method requires time to find the right number of cluster centers for the best monitoring performance. K-means RKPCA offers notable advantages, including enhanced computational efficiency for online monitoring, and robust capability to handle nonlinear dynamics. However, its primary disadvantages come from the need to carefully pre-define the number of clusters for K-means, which directly impacts the quality of the reduced data. This selection is crucial because the K-means itself can be sensitive to initial centroid placement and may converge to local optima, meaning that the chosen cluster representatives might not perfectly capture the underlying data structure.

The PCA-based RKPCA [36] uses PCA to select some uncorrelated observations. PCA is applied to the transpose of the training data set so that only uncorrelated samples are kept. PCA-based KPCA offers an intriguing approach to sample reduction, allowing for the identification of representative samples by capturing the dominant variations across observations. However, this method faces significant drawbacks. It can be computationally intensive for very large datasets due to the need to compute a very large covariance matrix, and its interpretation can be less intuitive, as the resulting components describe relationships between samples rather than features.

Spectral Clustering RKPCA and Random Sampling RKPCA in [42], the Spectral Clustering RKPCA uses spectral clustering to group data points based on the underlying structure of the point which is then used to define the number of clusters for the k-means clusters to form the reduced matrix. This reduction method has high time and storage complexities and scalability issues for large-sized data. Random Sampling RKPCA selects randomly some samples from the original data set to form the reduced matrix, this method is easy to use without any complicated steps but it is time-consuming and has a very large number of reduced matrices to choose from.

Authors in [43] presented an RKPCA algorithm that selectively retains a reduced number of observations. This method identifies samples whose transformed representation in the feature space exhibits a high projection value onto one PC rather than others using a given threshold. This method reduces the computational burden of standard KPCA by effectively removing less informative data. However, the efficacy of this reduction method is dependent on the appropriate choice of the number of principal components retained and the careful adjustment of the threshold, which directly influences the selection of representative samples.

Feature Vector Selection RKPCA [44] proposes a feature vector selection scheme based on geometrical considerations from [45] to reduce the computational complexity of KPCA when dealing with large training datasets. The core idea is to identify a minimal subset of samples whose mappings in the high-dimensional feature space can linearly express the entire dataset. This approach offers the significant advantage of directly addressing KPCA's computational burden by reducing the number of samples processed, thereby making it more feasible for large-scale applications while aiming to maintain data representativeness through its geometric selection principle. However, it comes with the disadvantage that the feature vector selection scheme itself can be computationally complex in its selection process, and there's an inherent risk of information loss if the chosen subset isn't perfectly representative, potentially impacting the accuracy of the resulting KPCA model.

3.3 Correlation Dimension RKPCA

3.3.1 Chaos Theory and Fractal Analysis

Chaos theory is a field of study interested in the qualitative behavior of deterministic nonlinear systems that exhibit unstable, non-periodic dynamics. While often counter-intuitive, chaos itself refers to the irregular, seemingly random behavior that can emerge from relatively simple governing equations, a phenomenon observed in various real-world domains [46, 47]. The butterfly effect, a core principle of chaos theory, states that even tiny initial changes can result in vastly different long-term outcomes [46]. Fundamentally, chaotic systems embody a delicate balance of order and unpredictability: they strictly adhere to underlying rules, yet their extreme sensitivity to starting conditions renders their long-term evolution practically unpredictable. This illustrates how seemingly random and complex behavior can arise from simple deterministic principles [48].
Fractals provide a unique descriptive framework for "wrinkled forms" that defy conventional Euclidean measures such as length or area, yet are distinctly not formless; they exist in a geometric "middle ground" [49]. Wrinkled forms describe shapes that are rough, jagged, and show the same intricate detail no matter how much zoomed in. The fundamental properties characterizing fractals are as follows:

- Fractal dimension: This non-integer dimension quantifies the complexity of a signal or shape and is a defining characteristic of fractals [49, 50]. Unlike integer dimensions that describe lines (1D) or planes (2D), fractal dimensions are typically represented by decimal values [46]. It is conceptually defined by the scaling relationship:

  $$bulk \quad \sim \quad size^{dimension} \tag{35}$$

  where *bulk* refers to a measure such as volume, mass, or information content, and *size* denotes a given linear distance. The symbol "$\sim$", in (35), denotes asymptotic proportionality. This means that as the scale s becomes infinitesimally small, the number of covering elements *bulk* becomes proportional to *size*, implying that the ratio between them approaches a non-zero constant value. Thus, the fractal *dimension* can be more formally expressed as:

  $$dimension \;=\; \lim_{size \to 0} \frac{log\,(bulk)}{log\,(size)} \tag{36}$$

  The use of the limit ensures invariance over smooth coordinate changes and establishes the dimension as a local quantity [49].

- Self-similarity: It is a key feature of fractals; they look like themselves at different scales. This can be exact, meaning a zoomed-in part is a perfect copy of the whole like the Koch snowflake or the Sierpinski gasket, or statistical, where the zoomed-in part shares the same overall characteristics like roughness or texture, but isn't an identical replica like coastlines, clouds, or mountain ranges.

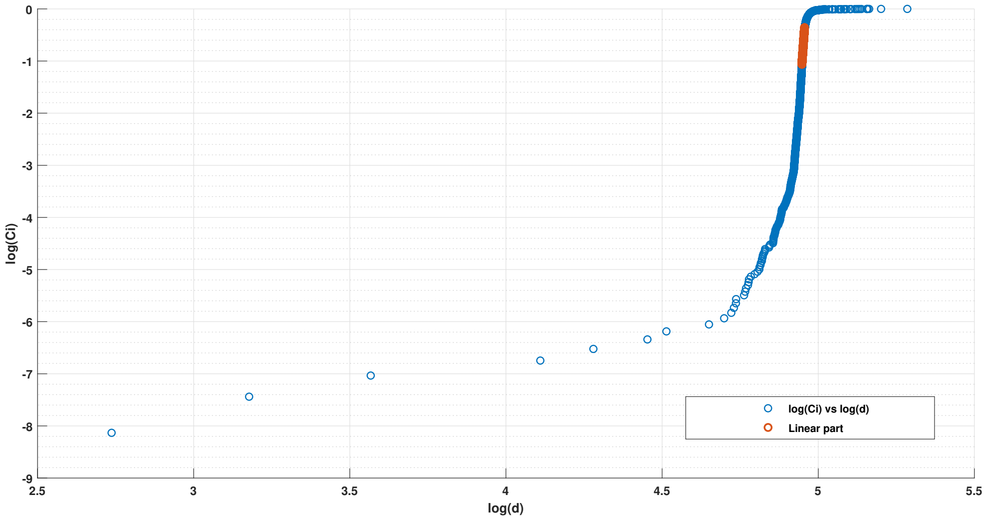

Figure 6: Example of CD Estimation of a given System.

Among the various methods for quantifying the complexity of fractal objects, such as the Hausdorff dimension, box-counting dimension, and self-similarity dimension, this thesis specifically utilizes the Correlation Dimension (CD). The CD provides a quantitative measure of self-similarity: a larger CD value indicates a higher degree of complexity and less self-similarity, and vice-versa [50]. The computation of CD relies on the Correlation Integral C_I, defined as:

$$C_I\left(n, d_i\right) = \frac{1}{n\left(n-1\right)} \sum_{i \neq j} \theta\left(d_i - \|X_i - X_j\|\right) \tag{37}$$

Here, θ represents the Heaviside function, and d is the radius of similarity. The values of d correspond to the Euclidean distances between rows (data points) in the original data set matrix. This equation can be conceptually simplified to:

$$C_I\left(n, d_i\right) = \frac{Number\ of\ distances < d_i}{\sum d} \tag{38}$$

Subsequently, a plot of $log(C_I)$ versus $log(d)$ is generated. The CD value is then deduced from the slope of the linear region within this plot. Figure 6 illustrates such a graph, with the linear portion, from which the slope (CD value) is computed, highlighted by red points.

3.3.2 Correlation Dimension RKPCA for FDD

As mentioned before, RKPCA algorithms consist of two major parts which are:

- Reduction part: this part is responsible for reducing the number of observations in the data set acquired from the monitored system by selecting only relevant observations using a given method.

- Build KPCA model: in this part, a KPCA model is built using the reduced data set obtained from the first part.

The Correlation Dimension RKPCA was used because CD measures the self-similarity of the system monitored, if the number of samples matches the CD ceiling value it can be said that these rows (samples) capture each correlation dimension and this reduced data might have sufficiently representative samples to analyze the system's fractal properties.

This algorithm is only used on chaotic systems, chaotic systems have a positive largest Lyapunov Exponent. The proposed reduction method used in this section is based on CD. It starts by computing the correlation dimension of the original data set, then omits the 1^{st} row of the original data set and computes the CD of the resulted matrix, if this CD value is the same as the CD value of the original data set then this row is omitted for good otherwise this row is kept in the reduced data, when this is done repeat the same process for the rest of rows in the original data set. In the end, the minimum retained observations have the same CD value as the original data set has, these observations form the reduced matrix used to build the KPCA model in the next part. Algorithm 5 illustrates how the proposed algorithm reduces the training data. The resulting reduced matrix X_r is then used to build the monitoring model based on algorithms 3 and 4.

Algorithm 5 Correlation Dimension RKPCA reduction part

```
Input: $X_o \in R^{n \times m}$
Output: $X_r \in R^{r \times m}$                                    ▷ $r < n$
Step 1: Standardize $X_o$
Normalize data with zero mean and unit variance using (8)
Step 2: Plot $log(C_I)\ vs\ log(d)$ and deduce CD
Obtain all Euclidean distances between every row vector $d$
Compute $C_I$ using (38)
Plot the $log(C_I)\ vs\ log(d)$ graph
Compute CD as the slope of the linear part of the graph
Step 3: Check for the right samples
while $i \leq n$ do
    Remove $i^{th}$ vector from training data set
    Compute CD of this matrix $CD_i$
    if $CD_i$ = CD then
        Remove this vector
        i= i+1
    else
        Keep this vector
        i= i+1
    end if
end while
Step 4: Form reduced matrix
The resulting matrix is now re-scaled using the inverse of (8) to form $X_r$
```

For Algorithm 5, the computational complexity is determined step-by-step. **Step 1**, which involves standardizing the $n \times m$ input data by calculating means and standard deviations for all m variables and then normalizing n samples, has a time complexity of $\mathcal{O}(n)$. **Step 2** begins by computing all pairwise euclidean distances among the n samples. As there are $\mathcal{O}(n^2)$ pairs, and each distance calculation for m-dimensional vectors takes $\mathcal{O}(m)$ operations, this part is $\mathcal{O}(n^2)$. Computing the Correlation Integral C_I is also dominated by this factor, making **Step 2**'s overall complexity $\mathcal{O}(n^2)$. The most computationally intensive part is **Step 3**, which contains a loop running n times. Inside this loop, the CD of a roughly $(n-1) \times m$ matrix is recomputed. Since each such re-computation has an $\mathcal{O}(n^2)$ complexity, the entire loop contributes $\mathcal{O}(n^3)$. Therefore, considering all steps, the overall time complexity of algorithm 5 is $\mathcal{O}(n^3)$.

3.4 Variogram-based RKPCA

3.4.1 Definition

Geo-statistics concerns the correlation between elements in a given time (and, or space) varying data set [51]. One of the geo-statistics techniques is the variogram, variogram quantifies spatial correlation and characterizes spatial continuity. The variogram model and empirical variogram are both parts of the variogram, the variogram model is the theoretical mathematical function that is fitted to the empirical variogram obtained from the experimental data set [52]. Characteristics of the variogram are summarized as the following in [51]:

- lag (h): It is the vector representing separation between two spatial locations.
- Nugget: it is the value of the variogram at $h = 0$.
- Range (a): The lag at which the variogram reaches the sill.
- Sill (c): It is the total variance of the data set, usually one for normal scores when the variogram is at the sill or close enough that there is no longer correlation between samples at that lag.

There exist different types of variogram models and here is some of the most used:

- Nugget: $\gamma_m(h) = \begin{cases} 0 & if \quad h = 0 \\ c & otherwise \end{cases}$.
- Spherical: $\gamma_m(h) = \begin{cases} c\left[1.5\left(\frac{h}{a}\right) - 0.5\left(\frac{h}{a}\right)^3\right] & if \quad h \leq a \\ c & otherwise \end{cases}$.
- Exponential: $\gamma_m(h) = c\left[1 - \exp\left(-3\frac{h}{a}\right)\right]$.
- Gaussian: $\gamma_m(h) = c\left[1 - \exp\left(-3\frac{h^2}{a^2}\right)\right]$.
- Power: $\gamma_m(h) = c.h^p, \quad 0 < p < 2$.

Empirical variogram, on the other hand, is a non-parametric estimator of the variogram of a spatial process [53]. The empirical variogram of the multivariate data set is then given as in (39) [54].

$$\gamma_j\left(h\right) = \frac{1}{2N\left(h\right)} \sum\nolimits_{i=1}^{N-h} \left(\bar{x}_{(i+h)j} - \bar{x}_{ij}\right)^2 \qquad i = 1 \cdots N-h, h = 1 \cdots N-1$$

$N\left(h\right)$ is the number of pairs that are separated by the lag h.

$$\gamma\left(h\right) = \frac{1}{m} \sum\nolimits_{j=1}^{m} \gamma_j\left(h\right) \tag{39}$$

γ_j is the univariate empirical variogram for each variable of the process and γ is the multivariate empirical variogram used for this study.

3.4.2 Variogram-based RKPCA for FDD

The Variogram-based RKPCA is proposed as a technique to eliminate correlated samples from the training dataset. This method uses the concept of a variogram, which is a tool in geostatistics that quantifies the spatial correlation between data points. Near the sill, samples that are separated by a specific lag distance and have a variogram value close enough to the sill are considered non-correlated samples and kept in the reduced matrix.

Let's define the Euclidean distance ω as the distance between the empirical variogram $\gamma\left(h\right)$ and its sill c. The proposed algorithm selects lags, h, for which the distance between $\gamma\left(h\right)$ and c is less or equal to ω. Then from the selected lags, the corresponding samples are kept to form the reduced matrix used to build the KPCA model. If the result is not satisfying then increase the value ω and repeat. Figure 7 illustrates what is the sill and the distance ω in a variogram plot.

Algorithm 6 is the algorithm used to reduce the original data set using empirical variogram. This algorithm may produce more than one reduced matrix, to select the appropriate one a monitoring model is built using one of the resulting X_r, algorithm 3, and algorithm 4 then the performance of this matrix is checked and a decision to use it or pass to the next one is made.

The time complexity of algorithm 6, which processes an input matrix X_o of n observations by m variables, is primarily determined by the variogram computation in **Step 2**. This step involves an outer loop iterating n times, within which a matrix subtraction and a subsequent matrix multiplication occur. This matrix multiplication dominates the inner loop, contributing $m^2(n-h)$. Summing this over the n outer iterations results in a complexity of $\mathcal{O}(n^2)$. While other steps like data standardization (**Step 1**) and reduced matrix formation (**Step 4**) involve complexities of $\mathcal{O}(n)$, these are generally overshadowed by the variogram calculation. Therefore, the overall time complexity of this algorithm is $\mathcal{O}(n^2)$.

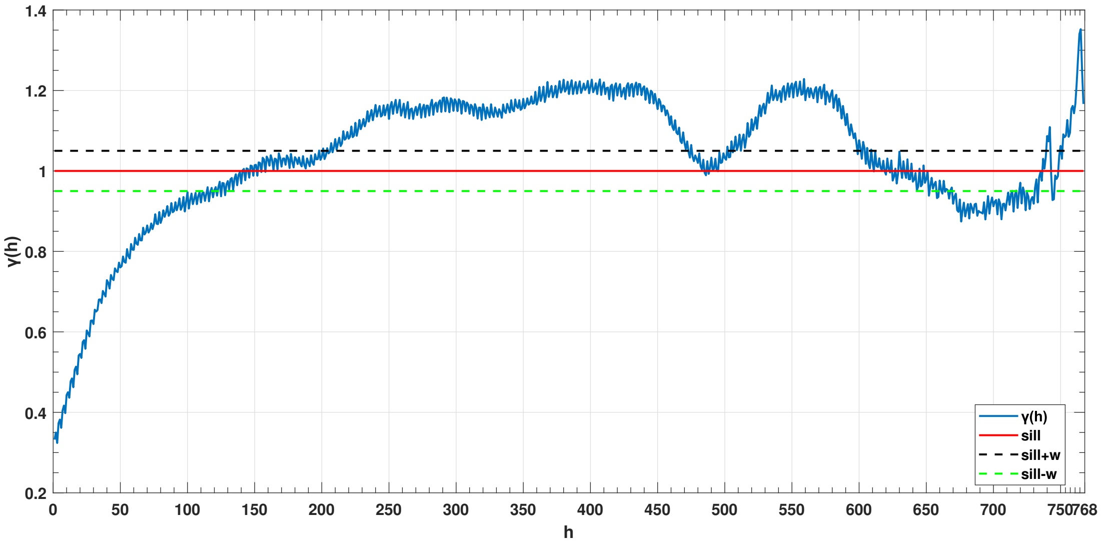

Figure 7: Empirical Variogram of CP Training Data.

Algorithm 6 Variogram-based RKPCA reduction part

```
Input: $X_o \in R^{n\times m}$, $\omega$
Output: $X_r \in R^{r\times m}$                    ▷ $r < n$ & they usually are more than one matrix
Step 1: Standardize $X_o$
Normalize data with zero mean and unit variance using (8)
Step 2: Compute variogram $\gamma(h)$
for $h = 1 \rightarrow n-1$ do
    $n(h) = n - h$
    $V = \left[\bar{X}_{(1:n-h)\times m} - \bar{X}_{(h+1:n)\times m}\right]^T \left[\bar{X}_{(1:n-h)\times m} - \bar{X}_{(h+1:n)\times m}\right]$
    $\gamma_j\,(h) = \frac{1}{2n(h)} V$                                        ▷ $j = 1 \;\cdots\; m$
    $\gamma(h) = mean(\gamma_j(h))$
end for
Step3: Compute different valuse of $\omega$
$v$ is a vector of all Euclidean distances between $\gamma(h)$ and $c$
Step 4: Form the reduced matrix for each $\omega$
for $\omega = min(v) \rightarrow max(v)$ do
    if $|\gamma(h) - c| \leq \omega$ then
        $s = min(h)$                                   ▷ $s$ is the minimum selected lag
        $X_r = \begin{bmatrix} X_{(1:n-s)\times m} \\ \bar{X}_{(s+1:n)\times m} \end{bmatrix}$
        The resulting matrix is now re-scaled using the inverse of (8)
    else
        go for higher value of $\omega$
    end if
end for
Step 5: Store all reduced matrices
```

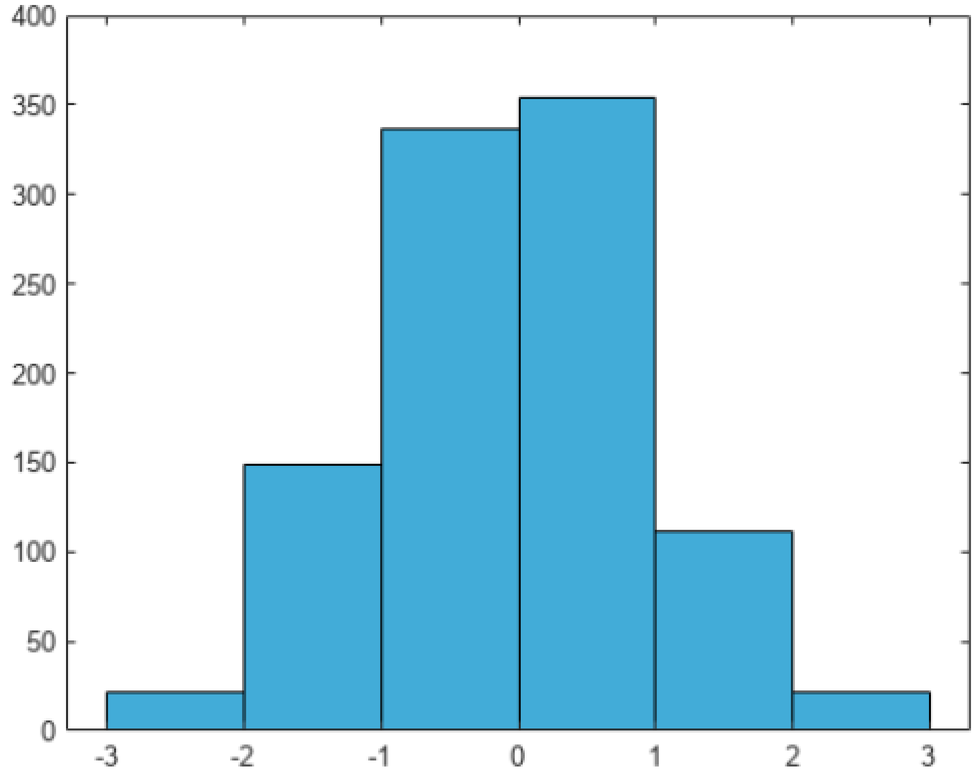

Figure 8: Histogram of the 1^{st} PC Score.

3.5 Histogram-based RKPCA

3.5.1 Definition

A histogram is a type of bar graph which displays the value of appearance frequency of specific data within a given bin, so it helps to visualize data distribution and its skewness, in other words, it visualizes how the data set is distributed [55]. The x-axis of the histogram represents equally divided intervals from the input data set known as bins and the width of the bins is controlled by the number of bins used as shown by the following equation:

$$B_w = \frac{M_V - m_V}{N_B} \tag{40}$$

where B_w is the bin width, M_V is the maximum value in data set, m_V is the minimum value in data set, and N_B is the number of bins. The y-axis can represent different types of values, such as appearance frequencies, probabilities, and percentages [55].

Figure 8 demonstrates an example of histogram plot, the y-axis represents the appearance frequency of the total values from each bin, the x-axis contains 6 bins with the width of 1, for this example. As can be seen from this figure the histogram can manifest the distribution of the data set (univariate data set).

3.5.2 Histogram-based RKPCA for FDD

The Histogram-based RKPCA is proposed because the histogram highlights the distribution of the data, and the PCA's first pincipal components score contains the highest percentage of the training data variations. Hence, using this algorithm ensures the same distribution in the reduced matrix as the original data set in the direction of the highest variations direction. The Histogram-based RKPCA starts by computing the 1^{st} PC score and plots its histogram with a specified number of bins N_B.

Lets define ε as the minimum appearance frequency of all bins in the histogram plot, ν_i as the appearance frequency of the i^{th} bin, and β_i as in (41).

$$\beta_i = \left\lceil \frac{\nu_i}{\varepsilon} \right\rceil \quad i = 1 \cdots N_B \tag{41}$$

For the i^{th} bin, it selects the number of observations equal to β_i such that the same median of the bin is kept. After this, the proposed algorithm selects the corresponding rows in the scores matrix and performs the inverse of PCA mapping to obtain the reduced matrix X_r that has less number of observations than the original data set.

Finally, the obtained matrices are used to build the KPCA model, using algorithms 3 and 4. The selected reduced matrix is the one with the satisfying results. The proposed reduction method is summarised in algorithm 7.

The complexity is driven primarily by **Step 2**, which performs PCA. Calculating the covariance matrix is $\mathcal{O}(m^2)$ as it is an $m \times m$ matrix, and performing eigenvalue decomposition on the resulting $m \times m$ covariance matrix is $\mathcal{O}(m^3)$. The step of computing the score matrix is $\mathcal{O}(m^2)$. These steps combine to give the dominant term. Subsequent steps, such as histogram creation and median computations (**Step 3**) and forming the reduced matrix (**Step 4**), have lower complexities relative to these PCA-related operations. Hence, the total time complexity for Histogram-based RKPCA is $\mathcal{O}(m^3)$

3.6 Homogeneity Testing and Divergence Estimation

A homogeneity test is a statistical hypothesis test used to determine whether two or more independent populations or groups share the same distribution [56]. The homogeneity test can be decomposed into the following:

- Purpose: The primary goal is to assess whether the proportions of observations that fall into each category are consistent between different populations or subgroups [57].
- Null hypothesis H_0: This hypothesis states that the distribution of the variable is identical across all populations being compared, meaning the proportions for each variable are equal [58].
- Alternative hypothesis H_a: This hypothesis posits that the distributions differ, indicating that at least one population has a distinct distribution for the variable compared to the others [58].

Algorithm 7 Histogram-based RKPCA reduction part

```
**Input:** $X_o \in R^{n \times m}$, $N_B$
**Output:** $X_r \in R^{r \times m}$          ▷ $r < n$ & they usually are more than one matrix
**Step 1: Standardize $X_o$**
Normalize data with zero mean and unit variance using (8)
**Step 2: Compute the $1^{st}$ PC score vector**
Compute Covariance matrix as in algorithm 1
Compute & sorte eigenvalues and eigenvectors as in algorithm 1
Compute the score matrix $T$ as in algorithm 1
Pick the $1^{st}$ column vector from $T$
**Step 3: Plot histogram of the obtained vector**
Specify the number of bins $N_B$
Plot histogram with $N_B$ as number of bins
Pick the minimum appearance frequency $\varepsilon$
**for** $i = 1 \rightarrow N_B$ **do**
    Compute the median of the $i^{th}$ bin
    Compute $\beta_i$ as in (41)
    Select $\beta_i$ observations from this bin with same median
**end for**
**Step 4: Form the reduced matrix**
Select the corresponding rows from the scores matrix $T$
Select the corresponding samples from $\tilde{X}$
Rescale this matrix to obtain $X_r$
**Step 5: Save all the reduced matrices**
```

Homogeneous data generally means that the data sets behave similarly in a specific statistical way, such as having the same proportions, variance, or overall distribution shape for the variables being tested. However, within the scope of this study, homogeneity carries a more direct implication: because the second dataset is directly deduced or obtained from the first, demonstrating their homogeneity statistically confirms that both datasets unequivocally represent the same underlying system. This provides crucial validation for the consistency and representativeness of the derived data with the original system behavior.

The homogeneity test used in this thesis was introduced in [59], it is a hypothesis test based on divergence estimation. The choice of this test is particularly pertinent because of the potential variability in the distributional properties of the data sets. It is not always guaranteed that the original data set follows a multivariate or univariate normal distribution. Traditional parametric tests often assume such normality, which might not be applicable in all scenarios. Given this limitation, the use of a non-parametric divergence estimation approach is more suitable. Non-parametric methods do not rely on specific distributional assumptions and are therefore more flexible in handling data sets that might not conform to traditional parametric models. This approach ensures a more robust assessment of homogeneity between the two data sets, accommodating potential deviations from normality, and providing a more reliable evaluation of whether the reduced data set adequately represents the original one.

Let P denote the distribution of the original data set, and Q denote the distribution of the reduced data set. Let p and q represent the probability density functions corresponding to distributions P and Q, respectively. The divergence function f from P to Q is defined as:

$$D_f\left(P, Q\right) = \int f\left(\frac{p\left(y\right)}{q\left(y\right)}\right) dQ\left(y\right) = E_Q\left(f\left(\frac{p\left(Y\right)}{q\left(Y\right)}\right)\right) \tag{42}$$

f is a convex function applied to the ratio $r\left(x\right) = \frac{p(x)}{q(x)}$, $f\left(1\right)$ is the minimal value of f-divergence if and only if $P = Q$. The convex function used for this test is the asymmetric Kullback-Leibler divergence and it is given as:

$$f_{aKL}\left(x\right) = x\ log\left(x\right) \tag{43}$$

This function was selected due to its simplicity and suitability for the study's needs. In this particular analysis, it is sufficient to compute the divergence from P to Q, as calculating the divergence in the reverse direction, from Q to P, is not required. This choice simplifies the computation and focuses on assessing how well the reduced data set Q approximates the original data set P without the need for a more complex bidirectional analysis. The function f_{aKL} has the following characteristic:

$$f_{aKL}\left(1\right) = 0$$

The main steps of the divergence estimation are as follows.

- Estimate the ratio $r\left(x\right) = \frac{p(x)}{q(x)}$ by $\hat{r}$.

- Estimate the divergence given f and $\hat{r}$.

The following algorithm 8 explains how these steps are performed.

Algorithm 8 Divergence estimation

```
**Step 1: Compute the kernel density estimation**
Estimate $\hat{p}$ and $\hat{q}$ using kernel density then set $\hat{r} = \frac{\hat{p}}{\hat{q}}$.
**Step 2: Smooth $f_{aKL}$**
Smooth the function $f_{aKL}\left(\hat{r}\left(x\right)\right).\hat{q}\left(x\right)$ via cubic splines.
**Step 3: integrate splines analytically**
```

For the hypothesis test, the null hypothesis is given as $H_0 : \quad P = Q$. At first from the matrix X_T, such that $X_T = \begin{bmatrix} X_o \\ X_r \end{bmatrix}$, and then from this matrix select n random observations to form one matrix and use the remaining observations to form another matrix, after that compute divergence between those obtained matrices using algorithm 8 and repeat these steps for b_t times when this is done there will be the total of $b_t + 1$ divergence estimates. The null hypothesis is true if the divergence on the original data sets exceeds the empirical one. the following equation explains how to compute the quantile of $b_t + 1$ estimations.

$$(1 - \alpha_t) - \ \mathit{quantile}\left(b_t + 1\right)$$

b_t and α_t are set by the user α_t is the significance level usually it is 5%.

3.7 Conclusion

In this chapter, three proposed algorithms for fault detection were introduced, each accompanied by a general description of their underlying concepts and their application. Each algorithm employs a distinct approach for determining how to retain essential information from the original data set in the reduced data set. The Correlation Dimension RKPCA algorithm is characterized by its method of consistently generating a single reduced matrix, regardless of the data characteristics. In contrast, the other two algorithms can produce multiple reduced matrices, from which the most effective one is selected based on performance metrics. A non-parametric homogeneity test is utilised to evaluate whether the reduced data set and the original data set adequately represent the same process. This test assesses whether the reduced data maintains the same statistical properties as the original data, providing a robust measure of the representational consistency between the two data sets.

4 Applications, Results and Discussion

4.1 Introduction

This chapter introduces the Tennessee Eastman Process (TEP) and the Cement Plant (CP) as benchmark examples. These well-established industrial processes serve as reliable testbeds for evaluating the proposed algorithms, providing a robust basis for assessing their performance in practical scenarios. Following their application to these two processes, a critical comparative analysis is presented. The performance of the proposed algorithms is systematically evaluated against established methods, focusing on key metrics such as overall monitoring effectiveness, execution time, required storage space, and homogeneity with the original dataset. The results are then discussed in detail.

The used software for this study is MATLAB and the hardware used has the following characteristics: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz and 8.00 Go of RAM with Windows 10 64 bits.

4.2 Tennessee Eastman Process

4.2.1 Tennessee Eastman Process Description

TEP developed by Downs and Vogel [60] in 1993, has become a widely used benchmark platform for evaluating fault diagnosis and process control algorithms. The TEP involves eight key components: G and H are the primary products; A, C, D, and E are reactants; F is a by-product; and B is an inert component. The process is structured around five main units: a reactor, a condenser, a recycle compressor, a separator, and a stripper. Figure 9 provides a general diagram of this process, illustrating the flow and interaction of the components and units within the system.

The reactions that happen inside the reactors are

$$A\,(g) + C\,(g) + D\,(g) \longrightarrow G\,(l)$$

$$A\,(g) + C\,(g) + E\,(g) \longrightarrow H\,(l)$$

$$A\,(g) + E\,(g) \longrightarrow G\,(l)$$

$$3D\,(g) \longrightarrow 2F\,(l)$$

g denotes gas and l is for liquide.

The TEP benchmark simulates 21 different faults which are presented in table 2, F 03, F 09, and F 15 faults are notoriously known for their hard detection [42].

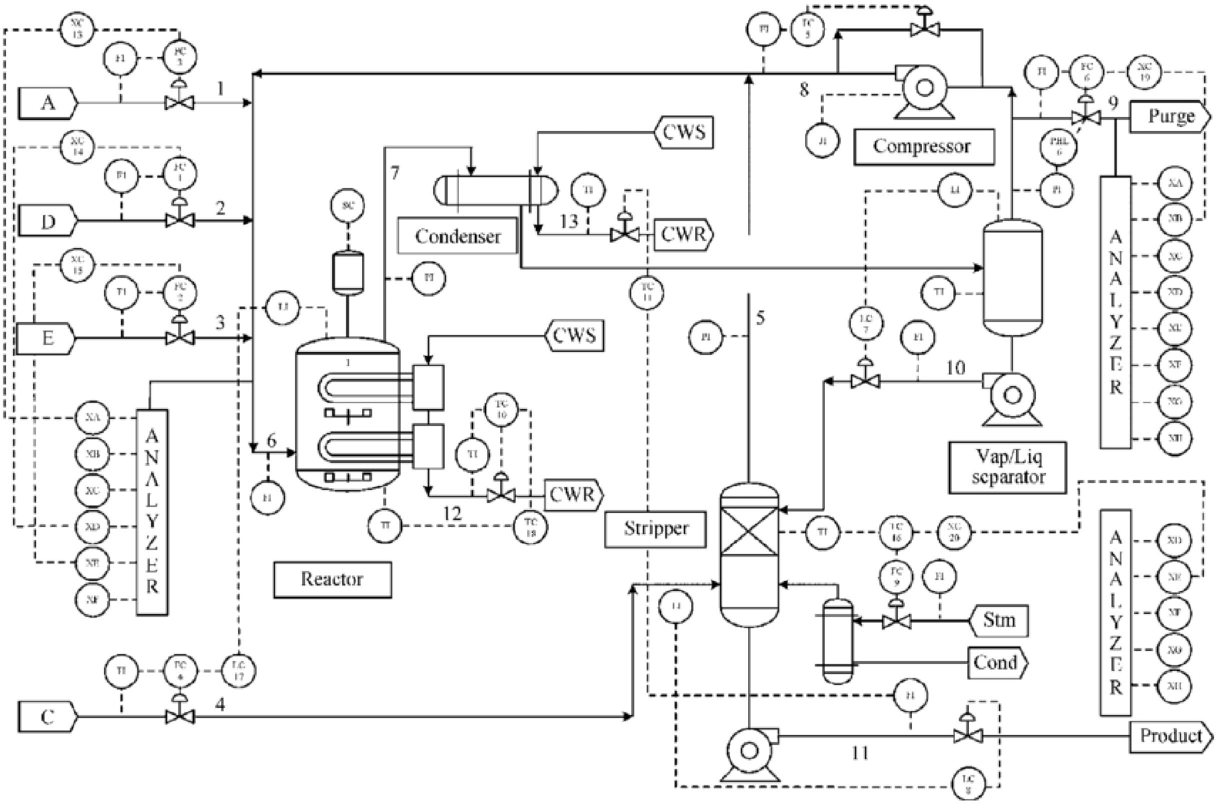

Figure 9: TEP Benchmark Process.

Table 2: Description of Different Faults of the TEP benchmark

Faults	Nature of fault	Description
F 01	Stepwise	A/C feed ration, B composition constant
F 02	Stepwise	B composition, A/C ratio constant
F 03	Stepwise	D feed temperature
F 04	Stepwise	Reactor cooling water inlet temperature
F 05	Stepwise	Condenser cooling water inlet temperature
F 06	Stepwise	A feed loss
F 07	Stepwise	C header pressure loss, reduced availability
F 08	Increase in variability	A, B, C feed composition
F 09	Increase in variability	D feed temperature
F 10	Increase in variability	C feed temperature
F 11	Increase in variability	Reactor cooling water inlet temperature
F 12	Increase in variability	Condenser cooling water inlet temperature
F 13	Slow driftwise	Reactor kinetics
F 14	Sticking valves	Reactor cooling water valve
F 15	Sticking valves	Condenser cooling water valve
F 16	Not Determined	Unknown
F 17	Not Determined	Unknown
F 18	Not Determined	Unknown
F 19	Not Determined	Unknown
F 20	Not Determined	Unknown
F 21	Sticking valves	The valve was fixed at the steady state position

4.2.2 Application Using KPCA

Using the "d00" training dataset from the TEP benchmark, a monitoring model for fault detection is constructed. This model employs algorithm 3 for the offline part of model building and algorithm 4 for the online monitoring part. The hyperparameter for the RBF kernel is chosen empirically for each monitoring index to optimize performance across all indices. Additionally, the significance level for the upper control limit of the monitoring indices is set at $\alpha = 99\%$.

Table 3 shows the monitoring results obtained using the conventional KPCA for different faulty scenarios. From this table, it can be noticed that KPCA has successfully detected the majority of faults with respectable monitoring performances for different monitoring indices and this was obtained by selecting the appropriate number of PC for each index, the proper number of PCs for the T^2 index is 51, for the Q index it is 36, and for the combined index φ it is 42. One can conclude from this that it is important to choose the appropriate number of retained PCs for the monitoring model. The φ index monitoring performances outperform the other two indices with the smallest margin followed by the Q index and then finally the T^2 index. Unfortunately, some faults are notoriously known for their hard detection like F 03, F09, and F 15 for all monitoring indices.

Table 3: KPCA Monitoring Results for TEP.

Indices		T^2			Q			φ	
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	3.75	0.13	1	10	0.13	1	3.13	0.25	1
F 02	1.88	1.13	8	10.63	0.75	0	1.88	1.25	10
F 03	14.38	76.13	3	16.88	75.00	1	21.25	73.25	3
F 04	3.75	0.00	0	10	0.00	0	2.50	17.88	0
F 05	3.75	60.25	0	10	59.38	0	2.50	56.00	0
F 06	0.00	0.00	0	6.25	0.00	0	91.25	0.00	0
F 07	0.00	0.00	0	5.00	0.00	0	5.00	0.00	0
F 08	6.25	0.75	6	12.50	1.00	6	10.00	0.63	0
F 09	31.25	79.13	0	29.38	79.25	0	35.63	74.75	0
F 10	3.75	10.38	7	17.50	20.25	0	5.00	13.13	5
F 11	5.00	17.25	1	14.38	15.88	0	5.63	27.13	5
F 12	16.25	0.25	2	19.38	0.25	2	24.38	0.13	1
F 13	1.88	4.13	7	8.75	3.88	1	1.88	3.75	7
F 14	4.38	0.00	0	13.13	0.00	0	3.13	0.00	0
F 15	0.63	74.13	91	8.13	72.13	3	1.25	73.63	91
F 16	43.75	5.63	0	26.25	22.13	4	53.75	7.75	0
F 17	4.38	3.63	19	16.88	2.38	10	3.13	5.38	21
F 18	3.13	8.13	14	16.88	7.63	15	11.25	6.13	3
F 19	1.25	15.50	1	8.75	41.25	1	2.50	20.63	0
F 20	1.25	22.63	67	5.00	22.50	5	0.00	21.75	5
F 21	11.25	45.63	1	21.88	32.75	0	12.50	53.38	48

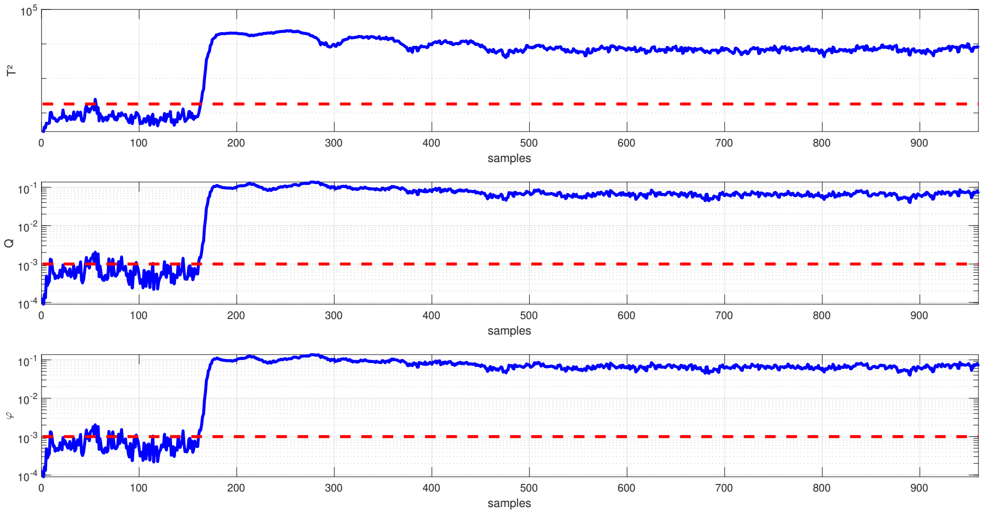

Figure 10: KPCA Monitoring for F 01.

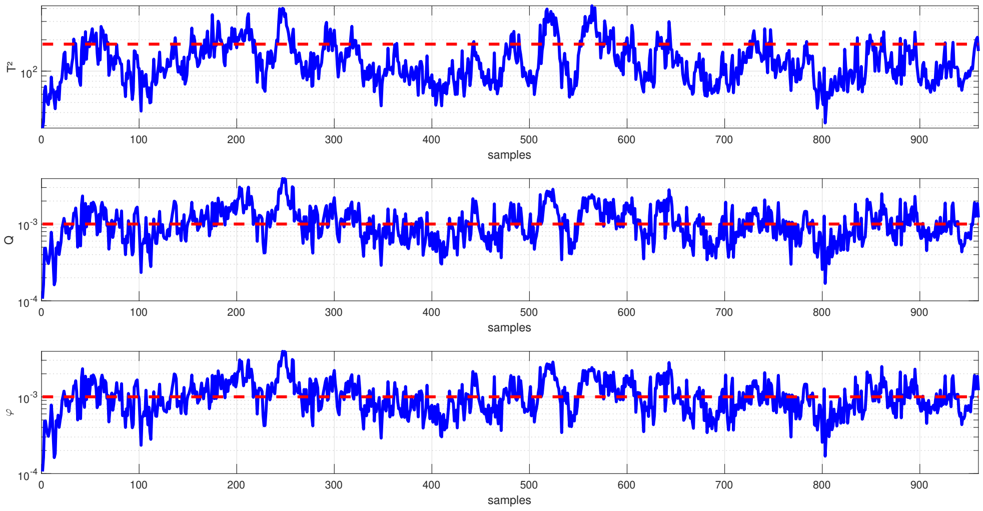

Figure 11: KPCA Monitoring for F 03.

Figure 10 shows the monitoring performance using the KPCA algorithm for F 01 for different monitoring indices as it can be noticed that this fault is successfully detected. The red line is the UCL limit for different monitoring indices. Figure 14 is the result of using the KPCA algorithm for the F 03 fault, this fault is hard to detect as seen in the same figure.

4.2.3 Application Using Correlation Dimension RKPCA

To apply a chaos theory approach, the first step is to assess whether the system in question is chaotic. This assessment involves calculating the largest Lyapunov Exponent from the dataset generated by the system. The Lyapunov Exponent quantifies the rate at which nearby trajectories in the system diverge over time. If the largest Lyapunov Exponent is positive, it indicates that the system is chaotic, as this positive value means that nearby trajectories are separated exponentially. This exponential separation of trajectories is a key characteristic of chaotic behavior[61]. For the TEP data set the largest Lyapunov exponent is equal to 4.31×10^{-4} which is a positive value and then the TEP is considered a chaotic system.

Then the graph of $log\,(C_I)\ \ vs\ log\,(d)$ is plotted as shown in figure 12, C_I is obtained using (38) and d is the euclidean distances between samples of the data set. The slop of the red part in figure 12 is equal to 20.95 which means that the correlation dimension value of the TEP training data set is 21 and the remaining samples from the reduction part should be also 21 samples.

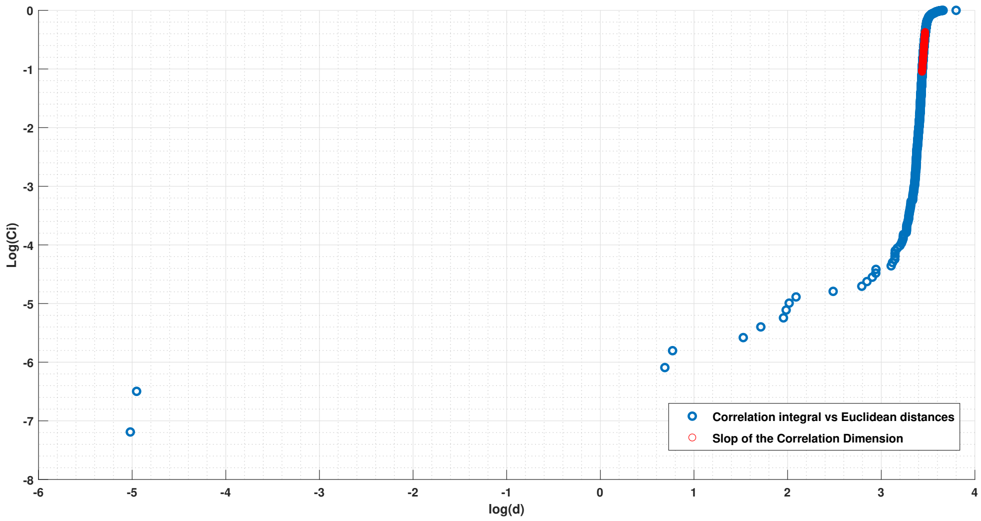

Figure 12: Estimation of CD (TEP).

After that algorithm 5 is applied to the training data set to reduce it, and the resulting matrix has only 21 samples, then use this matrix to build the monitoring model using algorithm 3. After that, algorithm 4 is used for monitoring the system under faulty scenarios. For the combined index, Correlation Dimension RKPCA uses a special formula to compute this index. φ index is given as

$$\varphi = (1 - \eta)\, T^2 + \eta Q \tag{44}$$

The value selected for η should lead to minimum variance in φ index, it can be calculated as

$$\eta = \frac{var\,(T^2)}{var\,(T^2) + var\,(Q)} \tag{45}$$

var in this equation is the variance of a given data set. For TEP training data set η is very close to one which means that both Q and φ indices have the same monitoring performances which can be noticed in table 4.

Table 4: Correlation Dimension RKPCA Monitoring Results for TEP.

Indices	T^2			Q			φ		
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	0.00	10.13	12	15.63	1.00	1	15.63	1.00	1
F 02	0.00	5.63	43	16.25	1.38	1	16.25	1.38	1
F 03	0.00	100	NA	20.63	66.13	1	20.63	66.13	1
F 04	0.00	100	NA	15.00	71.63	0	15.00	71.63	0
F 05	0.00	92.38	28	15.00	54.88	0	15.00	54.88	0
F 06	0.00	3.00	24	9.38	0.25	1	9.38	0.25	1
F 07	0.00	81.88	6	15.00	0.13	0	15.00	0.13	0
F 08	0.00	50.13	64	17.50	3.75	15	17.50	3.75	15
F 09	0.00	100	NA	35.00	69.50	0	35.00	69.50	0
F 10	0.00	100	NA	12.50	40.50	2	12.50	40.50	2
F 11	0.00	100	NA	25.63	55.50	6	25.63	55.50	6
F 12	0.00	57.88	105	16.88	2.13	0	16.88	2.13	0
F 13	0.00	26.88	90	15.63	2.88	2	15.63	2.88	2
F 14	0.00	84.00	9	25.00	0.38	1	25.00	0.38	1
F 15	0.00	100	NA	11.88	72.13	1	11.88	72.13	1
F 16	0.00	100	NA	41.25	41.25	0	41.25	41.25	0
F 17	0.00	57.63	35	26.88	12.63	9	26.88	12.63	9
F 18	0.00	15.13	108	14.38	8.38	0	14.38	8.38	0
F 19	0.00	100	NA	17.50	67.38	0	17.50	67.38	0
F 20	0.00	100	NA	11.25	47.00	10	11.25	47.00	10
F 21	0.00	100	NA	11.88	39.88	19	11.88	39.88	19

From Table 4, it is evident that the Correlation Dimension RKPCA has struggled to detect most faults for the T^2 index. This limited performance is attributed to the small number of PCs selected for this application, which is only 8, in contrast to the larger number used by the conventional KPCA algorithm. For the Q index, the Correlation Dimension RKPCA shows improved performance compared to the T^2 index but still does not surpass the conventional KPCA. Notably, this algorithm does outperform conventional KPCA in detecting certain notoriously difficult faults, such as F03, F05, F09, and F15. Similarly, for the combined index φ, while the monitoring performance is

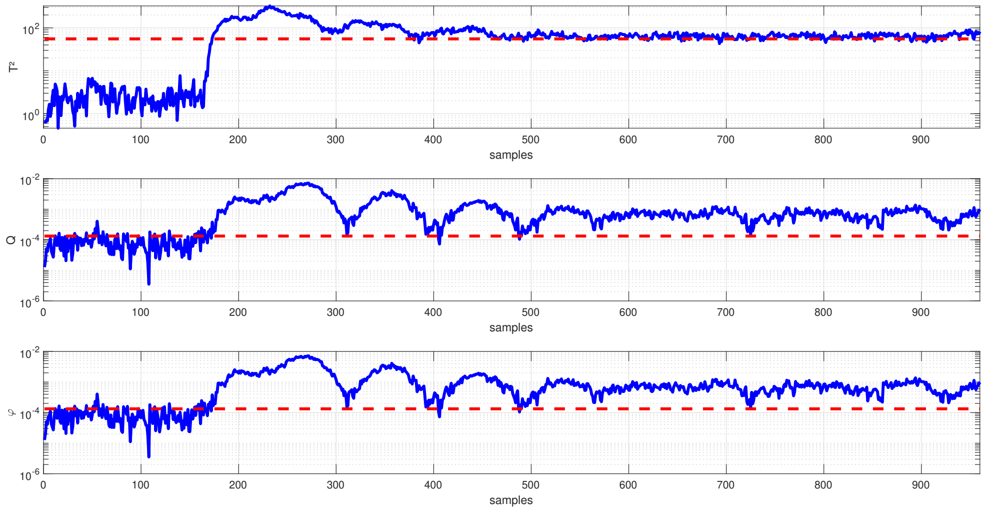

Figure 13: Correlation Dimension RKPCA Monitoring for F 01.

not exceptional, it is better than that of conventional KPCA for challenging faults. The Correlation Dimension RKPCA has significantly reduced the training dataset to just 21 samples, but this reduction negatively impacts the overall monitoring performance of the model.

Figure 13 shows the result of using the Correlation dimension RKPCA to monitor the first fault in the TEP benchmark, the fault has been successfully detected. Unlike the F 01 fault, F 03 is a hard-to-detect fault and the Correlation Dimension RKPCA failed to properly detect this fault as shown in figure 14.

4.2.4 Application Using Variogram-based RKPCA

The algorithm 6 is used to reduce the number of samples in the training data set using an empirical variogram. Figure 15 shows the variogram of the training data set, $\gamma\left(h\right)$, The red line is the sill of the variogram and the green lines, $sill \pm \omega$, are the borders used to select the appropriate lags. For the values of ω that are less than 2.2×10^{-6}, there are no selected $\gamma(h)$ and hence no selected lags; this means that the reduced matrix is empty. For the value of ω such that $2.2 \times 10^{-6} \leq \omega < 5.78 \times 10^{-4}$, the smallest lag, h, selected is $h = 264$, this lag results in a reduced matrix of 472 samples from the total of 500. Finally, for the values of ω for which $\omega > 5.78 \times 10^{-4}$ the smallest selected lag is $h = 108$, which results in a reduced matrix that is the same as the original.

The reduced matrix obtained using algorithm 6 and $2.2 \times 10^{-6} \leq \omega < 5.78 \times 10^{-4}$ is then used to build the monitoring model based on algorithm 3 with the appropriate number of PCs. Then this monitoring model is used in the online part by algorithm 4.

Table 5 presents the result obtained using the selected reduced matrix. The selected number of PCs for each monitoring index is 51 for the T^2 index, 120 for the Q index, and

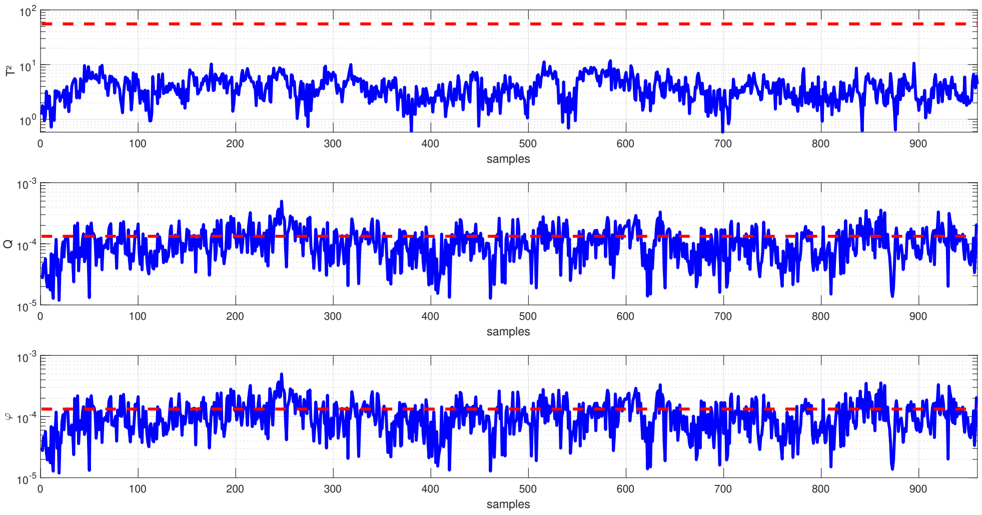

Figure 14: Correlation Dimension RKPCA Monitoring for F 03.

58 for the combined index φ. The proposed algorithm in this part performs as well as the KPCA algorithm for the T^2 index based on results obtained from tables 3 and 5. For the Q index, the proposed algorithm has slightly high FAR values and low MDR values this shows the trade-off relationship between monitoring metrics if one of them is low the other gets high. The combined index φ has successfully detected most of the faults. Generally, the Variogram-based RKPCA performs as anticipated for all monitoring indices unlike the Correlation Dimension RKPCA but the retain number of samples is quite high.

The variogram-based RKPCA has reduced the training data set by only 5.60% of the total number of samples because the Tennessee Eastman process data is a well-organized data set and it is quite challenging to reduce this data by a large amount without sacrificing the monitoring performances.

Figure 16 shows the monitoring performance using the Variogram-based RKPCA algorithm for F 01, in this figure the detection of the fault is clear and successful. Figure 17 shows how unsuccessful the model was in detecting the notorious F 03 fault.

4.2.5 Application Using Histogram-based RKPCA

The histogram-based RKPCA reduction part presented by algorithm 7 produces more than one reduced matrix and to choose which one is the appropriate a comparison between their performances is held to pick the right one. Before the comparison a monitoring model is build using algorithm 3 and then this model is used to monitor different faults by algorithm 4. Then, equation (6) is used to evaluate the total performance of a given model for a specified monitoring index.

Table 6 illustrates the performance of different monitoring models based on the cost function (6). For the T^2 index, the reduced matrix obtained using $N_B = 16$ has the best monitoring performance, this matrix has 256 samples followed by the one obtained using

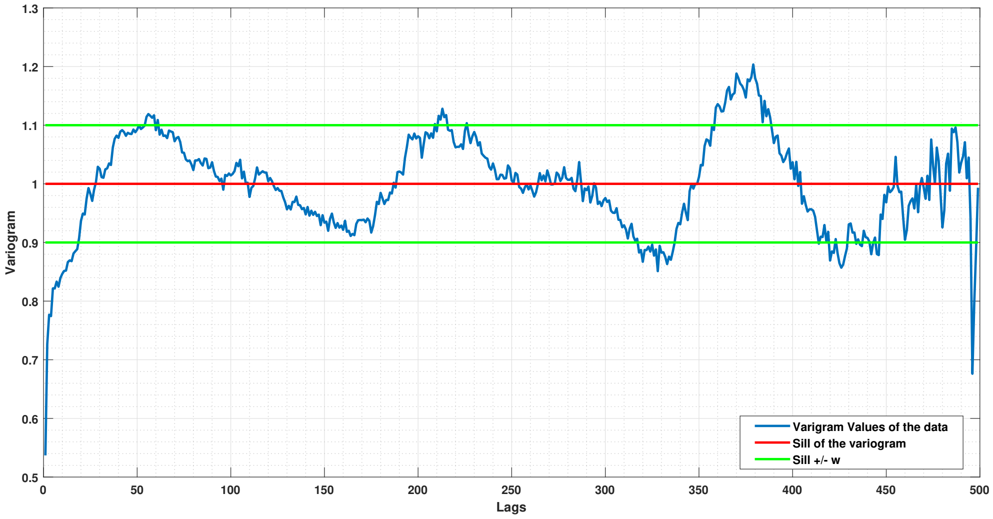

Figure 15: Empirical Variogram of TEP Training Data.

$N_B = 11$. $N_B = 18$ selects a reduced matrix with the best monitoring performance in terms of Q index. For the combined index φ, again $N_B = 16$ produces the matrix with the best monitoring performance. If the monitoring model is designed to focus on only one index, one should select the reduced matrix that corresponds to these optimal settings. It is important to note that matrices with the same number of retained samples might share the same minimum appearance frequency, but they are totally different.

In this study, the model is designed to monitor the process using all indices thus only one matrix is selected to build the model. Equation (7) is a cost function used for this purpose, it is the mean performance of all monitoring indices, $N_B = 16$ leads to the best overall performances.

Figure 18 is a histogram of the first principal component score for the original data set and figure 19 is one of the reduced data sets. Algorithm 7 has kept the same distribution of the data as it can be noticed from both histograms they are nearly identical with smaller appearance frequency.

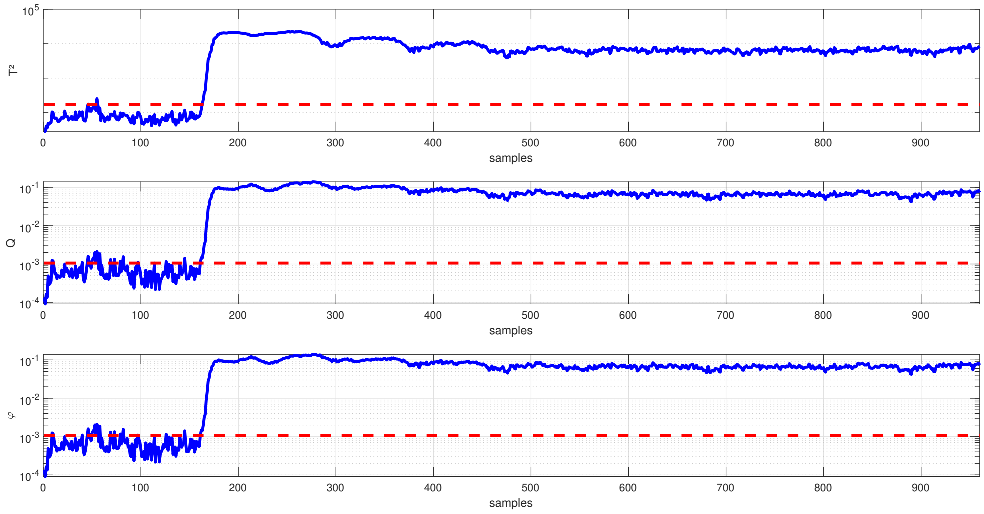

Figure 16: Variogram-based RKPCA Monitoring for F 01.

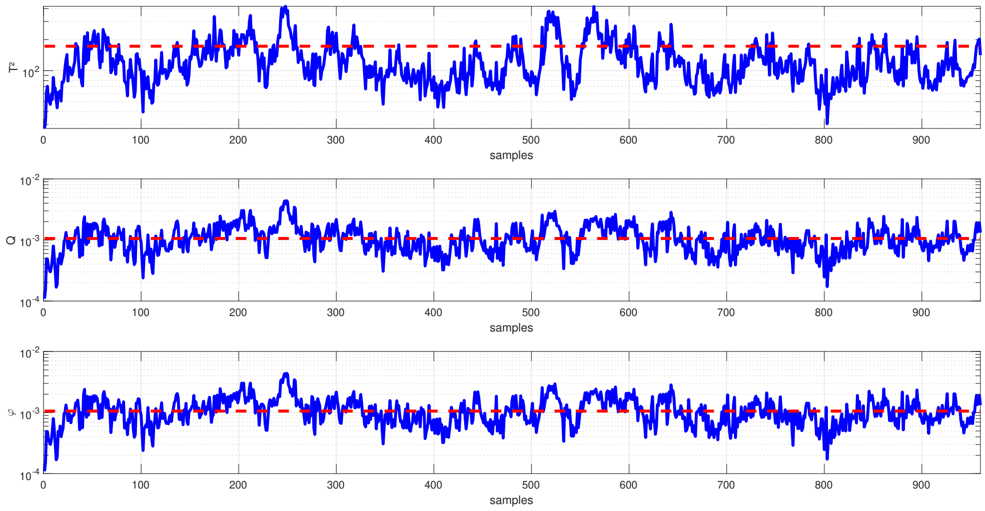

Figure 17: Correlation Dimension RKPCA Monitoring for F 03.

Table 5: Variogram RKPCA Monitoring Results for TEP.

Indices	T^2			Q			φ		
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	2.50	0.25	2	18.13	0.00	0	3.75	0.38	3
F 02	1.25	1.25	10	15.63	0.63	4	1.88	1.38	11
F 03	19.38	76.00	3	50	41.13	2	18.75	71.25	3
F 04	3.75	0.00	0	21.88	0.50	0	5.63	9.75	0
F 05	3.75	60.13	0	21.88	30.63	0	5.63	54.50	0
F 06	0.00	0.00	0	15.63	0.00	0	1.25	0.00	3
F 07	0.63	0.00	0	18.13	0.00	0	6.88	0.00	0
F 08	6.88	0.63	0	38.75	0.50	0	10.63	1.00	7
F 09	32.50	77.75	0	55.63	43.13	0	37.50	73.63	0
F 10	4.38	11.50	7	18.75	9.75	4	4.38	19.50	7
F 11	5.63	18.88	5	32.50	9.38	1	10	23.50	5
F 12	18.75	0.25	2	34.38	0.00	0	19.38	0.13	0
F 13	1.88	4.25	7	11.88	1.75	1	2.50	3.50	6
F 14	3.75	0.00	0	30.63	0.00	0	8.75	0..00	0
F 15	1.25	73.00	91	18.13	50.13	0	1.25	71.75	91
F 16	45.63	5.50	0	68.75	5.50	0	53.13	16.50	0
F 17	4.38	3.75	1	43.13	3.00	0	6.25	7.50	0
F 18	3.13	8.38	14	20	3.50	0	3.75	7.50	3
F 19	2.50	18.38	1	25.63	19.38	1	6.88	21.13	1
F 20	1.25	23.75	74	15.63	10.50	5	1.25	50.13	67
F 21	10.63	49.13	13	35.63	29.75	2	8.75	39.88	26

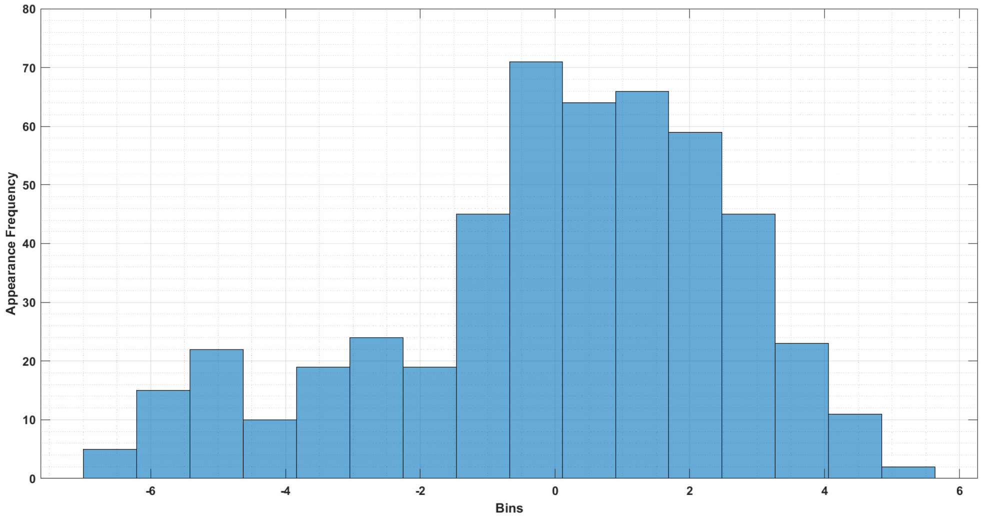

Figure 18: Histogram of the 1^{st} PC Score of Original Data (TEP).

Table 6: Monitoring Performances of Different Reduced Matrices.

		T^2		Q		φ	
N_B	ϵ	PC	J_s	PC	J_s	PC	J_s
19	02	48	0.67	38	0.48	41	0.46
18	02	41	0.76	39	**0.44**	43	0.44
17	02	46	0.67	38	0.54	43	0.44
16	02	46	**0.63**	37	0.50	41	**0.43**
15	02	51	0.70	36	0.49	41	0.47
14	04	46	0.88	41	0.50	17	0.56
13	05	42	0.92	15	0.56	21	0.53
11	02	45	0.65	36	0.52	36	0.51
10	03	48	0.69	41	0.48	20	0.50
09	12	20	1.26	13	0.64	13	0.56
08	12	19	1.28	13	0.64	13	0.58
07	21	10	1.49	11	0.65	14	0.67
06	11	29	1.34	14	0.63	13	0.60
05	24	8	1.53	18	0.66	11	0.96

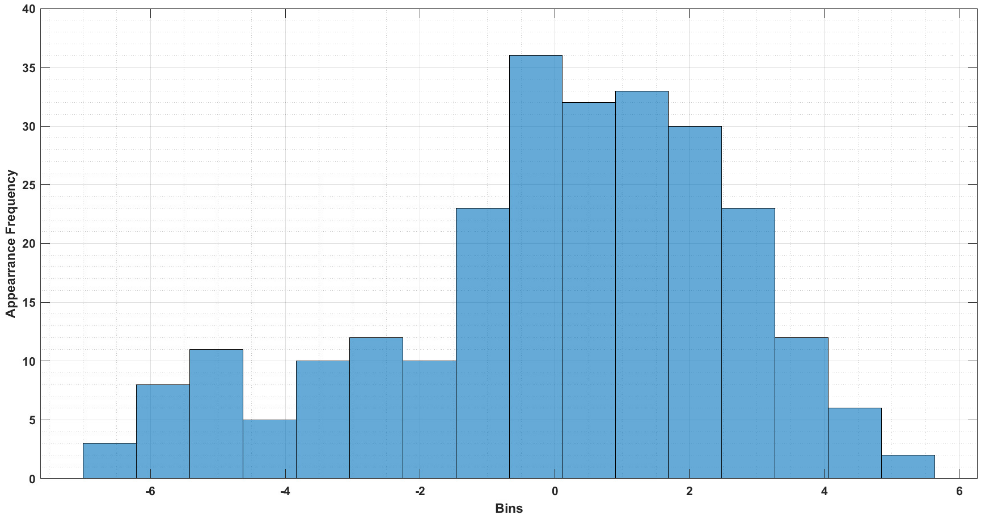

Figure 19: Histogram of the 1^{st} PC Score of Reduced Data (TEP).

N_B = 16 selects a reduced matrix of only 256 samples from the 500, table 7 presents the result obtained by using this reduced matrix to build the monitoring model. This model has successfully detected most faults, except the notorious ones, for all indices. Unfortunately, the monitoring model did not detect F 06 due to its value of FAR for the φ index.

The Histogram-based RKPCA has detected the F 01 fault without any challenges but it was not as successful in detecting the F 03 fault, figures 20 and 21 shows the monitoring results of the F 01 and F 03 faults, respectively. Histogram-based RKPCA generally shows increased FAR values for the Q index compared to KPCA results in table 3, but often

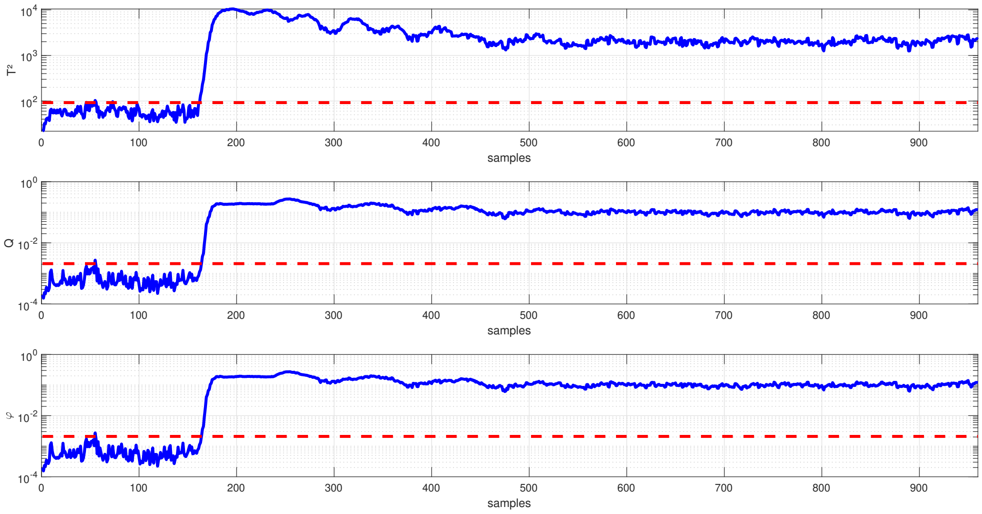

Figure 20: Histogram-based RKPCA Monitoring for F 01.

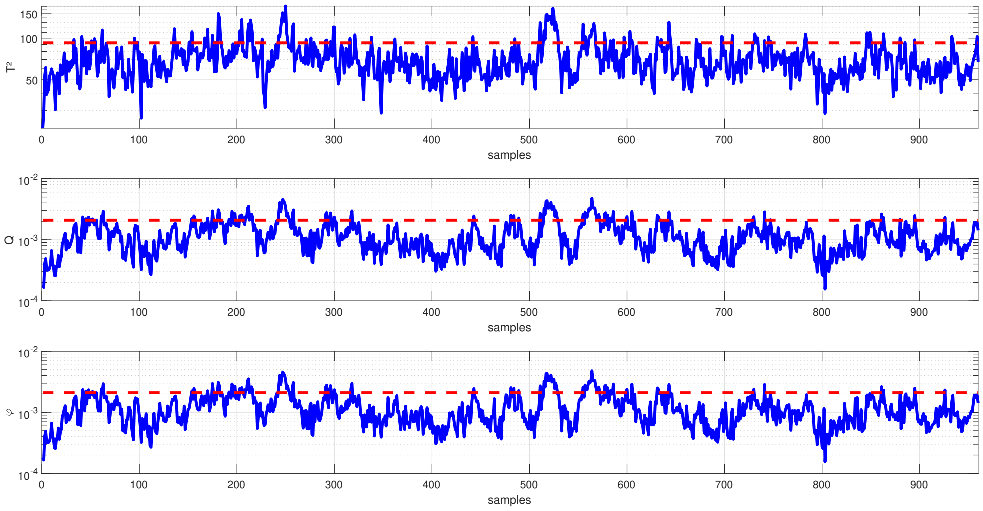

Figure 21: Histogram-based RKPCA Monitoring for F 03.

Table 7: Histogram-based RKPCA Monitoring Results for TEP.

Indices	T^2			Q			φ		
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	3.13	0.13	1	21.25	0.00	0	1.88	0.38	3
F 02	3.75	1.25	10	24.38	0.25	0	6.88	0.88	5
F 03	8.75	85.00	6	39.38	66.63	1	13.75	81.88	5
F 04	3.75	0.00	0	23.13	0.00	0	1.25	30.75	0
F 05	3.75	66.13	0	23.13	50.75	0	1.88	61.13	0
F 06	0.00	0.00	0	20.63	0.00	0	88.13	0.00	0
F 07	0.00	0.00	0	17.50	0.00	0	3.13	0.00	0
F 08	2.50	1.25	9	30.00	0.88	2	9.38	0.61	0
F 09	22.50	86.50	2	35.00	61.63	3	28.75	82	0
F 10	3.75	17.75	10	21.25	14.75	0	3.13	18.25	13
F 11	3.75	21.38	5	21.25	18.13	3	5.63	32.38	1
F 12	5.00	0.25	2	25.00	0.50	0	17.50	0.25	2
F 13	1.25	4.25	26	16.25	2.75	1	4.38	4.25	7
F 14	3.75	0.00	0	21.88	0.00	0	1.88	0.13	1
F 15	3.75	78.50	91	28.88	64.50	0	1.25	75.63	0
F 16	26.25	22.13	0	36.88	16.00	0	36.25	16.75	0
F 17	3.75	3.78	1	31.25	1.88	1	2.50	3.38	17
F 18	1.88	9.00	14	29.38	5.38	0	32.50	4.50	0
F 19	0.00	36.50	1	21.88	21.75	1	3.13	33.00	1
F 20	0.63	27.00	67	18.75	17.50	4	0.00	22.50	2
F 21	9.75	48.38	1	29.38	31.75	0	11.25	44.25	5

with improved (lower) MDR and DTD for specific difficult faults like F03, F09, and F15 (for Q) and F15 (for φ). For the T^2 index, table 7 often shows lower FAR values but at the cost of higher MDR values for difficult faults (F03, F09, F15) compared to table 3, making it less effective at detecting these specific faults.

4.2.6 Results and Discussion for Tennessee Eastman Process

Performance of monitoring systems built with a single monitoring index was evaluated by comparing proposed algorithms against conventional KPCA. A cost function J_s determined the best performing method. This comparison encompassed all matrices produced by the reduction methods, as well as the matrix from the KPCA algorithm. The results are compared based on tables 3, 4, 5, and 7.

For the T^2 index, The Variogram-based RKPCA algorithm exhibits the highest performance with a monitoring index value of $J_{T^2} = 0.56$. This indicates that the Variogram-based RKPCA method is the most effective at capturing and representing the variations and structure pertinent to the T^2 index. The conventional KPCA method ranks second with $J_{T^2} = 0.59$. While still demonstrating effective performance, it does not surpass the Variogram-based approach in terms of capturing the T^2 index variations. Following closely is the Histogram-based RKPCA with a reduced matrix obtained using $N_B = 16$, which achieves $J_{T^2} = 0.63$. This suggests that while the Histogram-based method is effective, it ranks below both the Variogram-based RKPCA and the conventional KPCA in terms of the T^2 index. The Correlation Dimension RKPCA shows the least favorable performance with $J_{T^2} = 1.64$. This significantly higher value indicates a poor monitoring performance, reflecting that this method is less capable of effectively capturing the features associated with the T^2 index. Therefore, when considering the T^2 index as the basis for monitoring, the Variogram-based RKPCA algorithm stands out as the most effective method for enhancing monitoring performance.

For the Q index, the Histogram-based RKPCA method, particularly with a reduced matrix obtained using $N_B = 18$, achieves the best result with $J_Q = 0.44$. This indicates that it provides the most effective monitoring performance among the methods tested concerning the Q index. The Variogram-based RKPCA follows with a Q index value of $J_Q = 0.50$. Although it shows strong performance, it is slightly less effective than the Histogram-based RKPCA in capturing the features pertinent to this index. The conventional KPCA method is next with $J_Q = 0.52$, reflecting a moderate level of performance that is surpassed by both Histogram-based and Variogram-based RKPCA approaches. The Correlation Dimension RKPCA ranks last with a Q index value of $J_Q = 0.70$, indicating that it performs the least effectively in capturing the relevant features for this index. Based the Q index, it shows that both Histogram-based RKPCA and Variogram-based RKPCA enhance the monitoring performance of the KPCA algorithm. In contrast, the Correlation Dimension RKPCA does not perform as well in this regard.

For the φ index, The Histogram-based RKPCA method, with a reduced matrix obtained using $N_B = 16$, achieves the best monitoring performance with $J_\varphi = 0.43$. This highlights its superior ability to capture the relevant features associated with this index. The conventional KPCA comes in second with $J_\varphi = 0.44$. Although it shows strong performance, it is slightly less effective compared to the Histogram-based RKPCA for this particular index. The Variogram-based RKPCA is third with a φ index value of $J_\varphi = 0.45$, demonstrating competent performance but trailing behind the Histogram-based KPCA and conventional KPCA. The Correlation Dimension RKPCA again shows the least favorable performance with $J_\varphi = 0.70$, indicating poor monitoring performance in comparison to the other methods. The Variogram-based RKPCA and Histogram-based RKPCA have enhanced the monitoring performances of the KPCA algorithm when using one index in the monitoring model.

Now it is the time to conduct a comprehensive comparison of the monitoring model's performance based on three different monitoring indices all at once. This comparison focuses on the value of the cost function J to assess and evaluate the effectiveness of each approach. To ensure a thorough analysis, we first selected the appropriate reduced matrices from each proposed approach and then compared their performance against several existing methods to determine if they have indeed enhanced performance.
Table 8 provides a detailed summary of the monitoring performances for each index based on the cost function defined in (6). This table includes several important metrics, including the number of retained samples in the reduced matrices. Notably, the Correlation Dimension RKPCA approach features the smallest number of samples in the training

dataset, followed by the PCA-based RKPCA with 44 samples. Conventional PCA comes next, and then the Histogram-based RKPCA, which impressively reduces the dataset by approximately 50% of the original number of samples. In contrast, the remaining RKPCA algorithms only manage to reduce the dataset by less than 6%. This variance in reduction underscores the need to compare the monitoring performances of these different approaches to evaluate their effectiveness. The KPCA and Variogram-based RKPCA approaches demonstrate the best monitoring performance with a cost function value of $J_{T^2} = 0.59$. This indicates that both methods effectively capture the necessary variations and provide strong performance based on the T^2 index. The Histogram-based RKPCA follows with a cost function value of $J_{T^2} = 0.63$, slightly trailing behind the top-performing methods but still performing well. The Euclidean Distance RKPCA and PCA algorithm exhibit comparable performance, which is less impressive compared to the top approaches. The Correlation Dimension RKPCA and PCA-based RKPCA bring up the rear with the lowest performance. From these results, it is evident that the Variogram-based RKPCA maintains the strong performance of the KPCA algorithm, while the Histogram-based RKPCA shows slightly reduced effectiveness. Both the Variogram-based and Histogram-based RKPCA methods achieve the best monitoring performance based on J_Q, significantly outperforming the KPCA algorithm. These approaches demonstrate considerable improvements in monitoring performance. Following these, the PCA and Euclidean Distance RKPCA methods show moderate performance levels. The Correlation Dimension RKPCA and PCA-based RKPCA again occupy the last positions. The results highlight that Variogram-based and Histogram-based RKPCA approaches effectively enhance the KPCA algorithm's monitoring capabilities based on the Q index, whereas the Correlation Dimension RKPCA does not perform as well. For the combined index φ, the KPCA, Euclidean Distance RKPCA, and Histogram-based RKPCA methods achieve the best monitoring values, reflecting strong overall performance. The Variogram-based RKPCA and PCA algorithm are positioned immediately following these top methods, indicating competitive performance.
The overall performance is summarized in Table 8, which calculates the mean of all monitoring results, as given in (7). This table allows users to identify which approach offers a balanced performance across all indices. In terms of overall performance, the Variogram-based RKPCA stands out with the best performance at $J = 0.51$, followed closely by both KPCA and Histogram-based RKPCA, each with a value of $J = 0.52$. The Euclidean Distance RKPCA holds the third spot with a value of $J = 0.58$, and the PCA algorithm ranks fourth with a value of $J = 0.59$. The Correlation Dimension RKPCA has an overall value of $J = 1.01$, and the PCA-based RKPCA is the lowest with a value of $J = 1.19$.
The results indicate that both Variogram-based and Histogram-based RKPCAs do not only preserve but often enhance the monitoring performance of the KPCA algorithm, even with a reduced number of samples. On the other hand, the Correlation Dimension RKPCA, while significantly reducing the number of samples, is only effective for specific types of faults and generally does not offer improved performance across the board.

After evaluating the monitoring performances of various approaches, the next step is to assess the homogeneity between the reduced matrices and the original one. To do this, the number of non-homogeneous variables is checked. This comparison is crucial as it helps determine how well the reduced matrices represent the original system.

Table 8: TEP Cost Function J-values for Different Algorithms.

		T^2		Q		φ	
Method	Size	PCs	J	PCs	J	PCs	J
PCA	52	41	0.76	35	0.55	41	0.47
KPCA	500	51	**0.59**	35	0.52	42	0.44
Euclidean Distance RKPCA [19]	497	42	0.75	35	0.56	41	0.44
Variogram-based RKPCA[62]	475	51	**0.59**	120	**0.50**	58	0.45
PCA-based RKPCA[36]	44	18	1.85	31	1.00	19	0.72
Correlation Dimension RKPCA[63]	21	08	1.64	14	0.70	14	0.70
Histogram-based RKPCA[64]	256	46	0.63	37	**0.50**	42	**0.43**

Table 9 provides a comprehensive overview of the number of non-homogeneous variables found between the reduced matrices and the original one. The results reveal that the Euclidean Distance RKPCA, Variogram-based RKPCA, and Histogram-based RKPCA approaches each have zero non-homogeneous variables. This indicates that these reduced matrices are perfectly aligned with the original data, maintaining the same system representation as the original dataset. In contrast, the Correlation Dimension RKPCA approach contains one non-homogeneous variable. Although this is a minor discrepancy, it is noteworthy considering that this approach has significantly reduced the size of the dataset by nearly 97%. The presence of a single non-homogeneous variable suggests that, despite the drastic reduction in dataset size, the reduced matrix still closely mirrors the original data's characteristics. On the other hand, the PCA-based RKPCA approach shows a much larger discrepancy, with a total of 32 non-homogeneous variables. This indicates a substantial deviation from the original dataset, reflecting that the reduced matrix does not adequately represent the original data. Such a high number of non-homogeneous variables suggests that the reduced matrix is significantly different from the original dataset, potentially affecting the effectiveness of any subsequent monitoring performed using this approach.

The findings in Table 9 are consistent with the results presented in Table 8. The homogeneous datasets—those with zero or minimal non-homogeneous variables—demonstrate strong monitoring performances. Conversely, the dataset with a higher number of non-homogeneous variables shows poorer monitoring performance. This alignment underscores the importance of maintaining data homogeneity to ensure effective monitoring and accurate representation of the original system.

The execution time taken to monitor one sample is directly related to the number of samples in the training data set as well as the required storage space for the monitoring model, figure 22 shows the relation between the execution time and the number of samples

Table 9: TEP non-homogeneous variables for different algorithms.

Method	Non-Homogeneous variables
Euclidean Distance RKPCA [19]	**0**
Variogram-based RKPCA [62]	**0**
PCA-based RKPCA [36]	$32 \sim [x1, \ldots, x9, x14, x21, \cdots$ $x29, x34, \ldots, x43, x45, \ldots x51]$
Correlation Dimension RKPCA [63]	$1 \sim [x16]$
Histogram-based RKPCA [64]	**0**

in the data set in terms of the T^2 index, it can be noticed that the required time always goes up for more samples in the training data set which is further explained by the following equation:

$$T^2 \to \begin{cases} E\left(n\right) = 7.591 \times 10^{-12} n^3 - 2.216 \times 10^{-9} n^2 + 8.606 \times 10^{-7} n + 2.773 \times 10^{-6}, & 0 \le n \le 361 \\ E\left(n\right) = -2.933 \times 10^{-10} n^3 + 3.934 \times 10^{-7} n^2 - 1.686 \times 10^{-4} n + 0.02429, & n > 361 \end{cases}$$

This equation is plotted as green and blue lines in figure 22 and the red diamond-shaped points are the measured values, the same thing goes for the other two monitoring indices they follow nearly the same behaviour and their fitting equations are given by:

$$Q \to \begin{cases} E\left(n\right) = 1.234 \times 10^{-11} n^3 - 3.042 \times 10^{-9} n^2 + 8.614 \times 10^{-7} n + 1.652 \times 10^{-6}, & 0 \le n \le 361 \\ E\left(n\right) = -4.741 \times 10^{-10} n^3 + 6.337 \times 10^{-7} n^2 - 2.748 \times 10^{-4} n + 0.04004, & n > 361 \end{cases}$$

$$\varphi \to \begin{cases} E\left(n\right) = 1.012 \times 10^{-11} n^3 - 1.797 \times 10^{-9} n^2 + 5.827 \times 10^{-7} n + 5.396 \times 10^{-6}, & 0 \le n \le 361 \\ E\left(n\right) = -1.778 \times 10^{-10} n^3 + 2.590 \times 10^{-7} n^2 - 1.175 \times 10^{-4} n + 0.01812, & n > 361 \end{cases}$$

These equations were derived by measuring the execution time and storage space for various indices. The execution time data was then fitted with a cubic polynomial, aligning with an expected $\mathcal{O}(n^3)$ time complexity. Similarly, the storage space measurements were fitted with a second-degree polynomial, consistent with an $\mathcal{O}(n^2)$ storage complexity. The split in the measurement for the execution time is produced by MATLAB software.

Table 10 contains the measured execution time, it is clear that the lower the number of samples the lower the execution time is and vice versa, the lowest execution time for each monitoring index goes to the Correlation Dimension RKPCA and the largest ones are those of the conventional KPCA. Figure 23 shows the required storage space for the monitoring model of all indices, the red circles refer to the measured storage space and the blue line is for the fitting equation which is:

$$S\left(n\right) = 1.436 \times 10^{-2} n^2 + 1.352 \times 10^{-1} n + 1.133$$

from both table 10 and figure 23 it can be concluded that the required storage space of the monitoring model is directly related to the number of samples in the training data set.

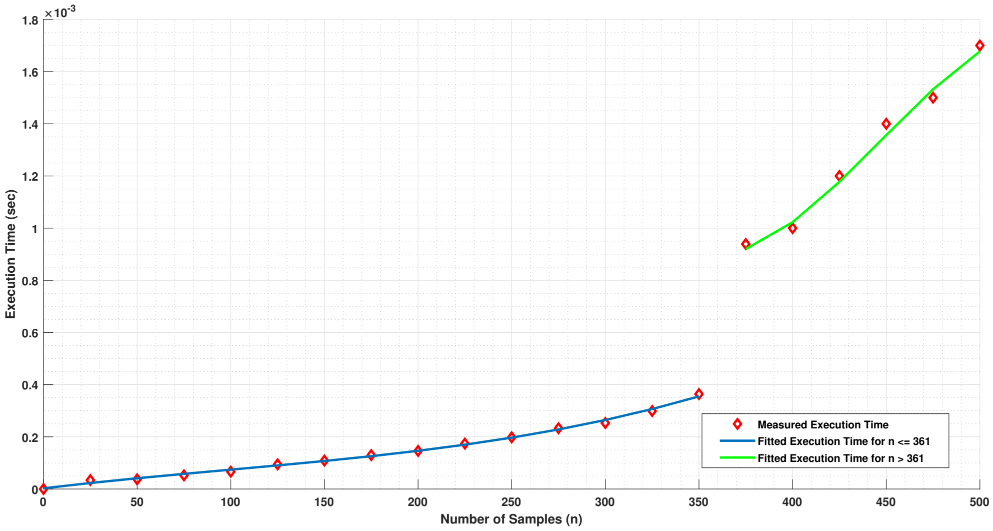

Figure 22: Execution Time for Different Number of Samples (T^2-index).

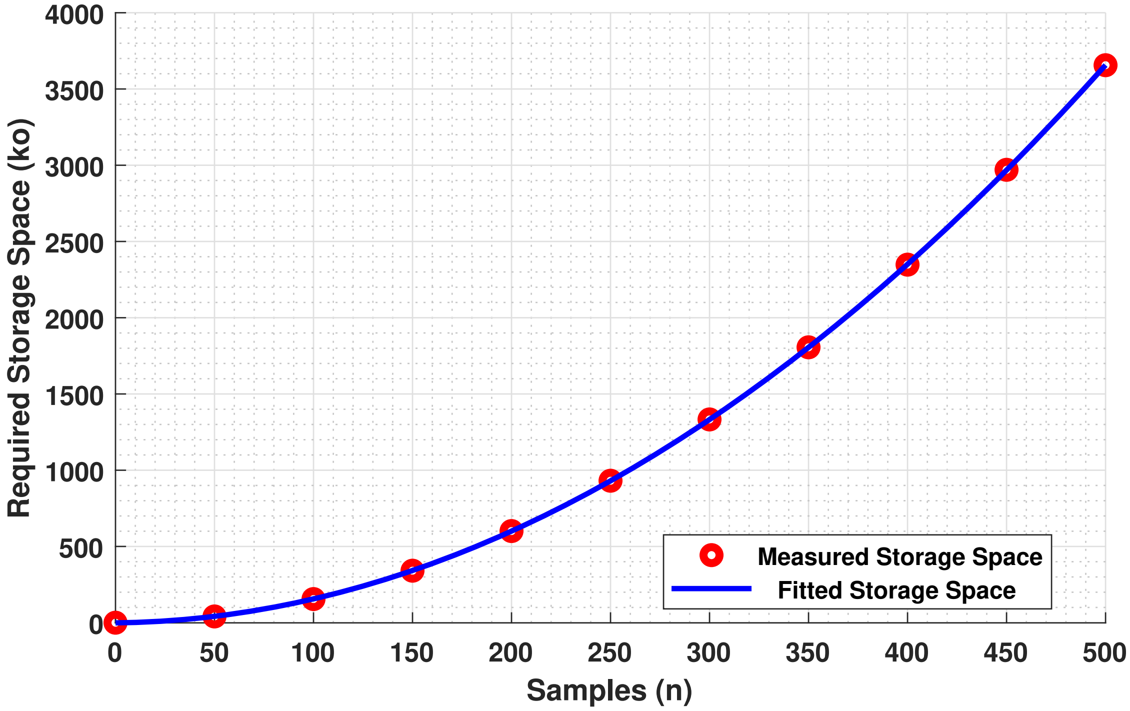

Figure 23: Required Storage Space per Number of Samples.

4.3 Cement Plant Rotary Kiln

4.3.1 Cement Plant Rotary Kiln Description

Cement production is a complex process that starts by mining and then grinding raw materials including limestone and clay to a fine powder, called raw meal, which is then

Table 10: TEP execution time and required storage space

			T^2	Q	φ
Method	Size	Storage Space (ko)	Execution Time (ms)	Execution Time (ms)	Execution Time (ms)
PCA	52	118	7.50×10^{-2}	6.83×10^{-2}	8.43×10^{-2}
KPCA	500	3822	1.70	1.80	1.90
Euclidean Distance RKPCA [19]	497	3765	1.70	1.80	1.80
Variogram-based RKPCA [62]	475	3417	1.50	1.70	1.70
PCA-based RKPCA [36]	44	41	3.59×10^{-2}	3.29×10^{-2}	3.81×10^{-2}
Correlation Dimension RKPCA [63]	21	11	2.96×10^{-2}	2.55×10^{-2}	2.55×10^{-2}
Histogram-based RKPCA [64]	256	1067	0.27	0.31	0.33

heated to a sintering temperature as high as 1450 ^oC in a cement kiln to broke the chemical bounds of the raw materials and then they are recombined to form new compounds. The result is called clinker, which is grounded to a fine powder in a cement mill and mixed with gypsum to create cement.

Ain El Kebira cement plant is located near Setif in the eastern of Algeria. It has a rotary kiln of 5.4 m shell diameter and 80 m length with 30^o incline. The kiln is spun up to 2.14 rpm using two 560 kws asynchronous motors and the producing clinker of density varying from 1300 to 1450 $kg.m^{-3}$ under normal conditions. Two natural gas burners are used, the main one in the discharge end and the other one in the first level of the preheater tower without any tertiary air conduct. The schematic diagram is presented in figure 24. A description of the different process variables is reported in the following table.

Table 11: Variables of the cement plant rotary kiln

Signal	Description	Unit
x_1, x_3, x_5, x_7	Depression of gases in outlets of cyclones (one, two, three, and four respectively) in tower I.	$mbar$
x_2, x_4, x_6, x_8	Temperature of gases in outlets of cyclones (one, two, three, and four respectively) in tower I.	$^\circ C$
x_{10}	Depression of gas in inlet of cyclone four tower I.	$mbar$
$x_{17}, x_{19}, x_{21}, x_{23}$	Depression of gases in outlets of cyclones (one, two, three, and four respectively) in tower II.	$mbar$
$x_{18}, x_{20}, x_{22}, x_{24}$	Temperature of gases in outlets of cyclones (one, two, three, and four respectively) in tower II.	$^\circ C$
x_{12}, x_{25}	Temperature of the material entering the kiln from tower I and tower II respectively.	$^\circ C$
x_9, x_{15}	Power of the motor driving the exhauster fans of tower I and tower II respectively.	kW
x_{11}, x_{16}	Speed of the exhauster fans of tower I and tower II respectively.	rpm
x_{13}	Depression of gas in the outlet of the smoke filter of tower I.	$mbar$
x_{14}, x_{26}	Temperature of gas in the outlet of the smoke filters of tower I and tower II respectively.	$^\circ\mathrm{C}$
x_{27}	The sum of the powers of the two motors spinning the kiln.	kW
x_{28}	Temperature of excess air from the cooler	$^\circ\mathrm{C}$
x_{31}	Temperature of the secondary air	$^\circ\mathrm{C}$
x_{29}, x_{32}, x_{33}	Pressure of air under static grille, repression of fan I, fan II, and fan III respectively.	$mbar$
x_{30}, x_{34}	Speed of the cooling fan I and fan III respectively.	rpm
x_{35}, x_{37}, x_{39}	Pressure of air under the chamber I, II, and III of the dynamic grille, repression of fan IV, V, and VI respectively.	$mbar$
x_{36}, x_{38}, x_{40}	Speed of cooling fan IV, V, and VI respectively.	rpm
x_{41}	Speed of the dynamic grille.	$Stroke\,\mathrm{min}$
x_{42}	Command issue of the pressure regulator for the speeds of the draft fans of cooler filter.	rpm
x_{43}	Flow of fuel (natural gas) to the main burner.	$m^3\,\mathrm{h}^{-1}$
x_{44}	Flow of fuel (natural gas) to the secondary burner (pre-calcination level).	$m^3\,\mathrm{h}^{-1}$

Data sets used for this work are:

- training data set contains 768 observation samples collected during normal condition operation, with a sampling rate of $20sec$.

- Testing data set with 11000 observations. This set was collected from the plant during healthy operation with a sampling interval of $1sec$.
- Real process fault with 2048 observations, where the fault increases gradually after 420 samples.
- 10 simulated sensor faults data sets with each one having 1000 observations.

Table 12 shows the simulated faults, their type, affected variables, fault magnitude, and samples affected by these faults.

Table 12: Introduced simulated faults

Name	Type	Affected Variables	Fault magnitude	Fault Range
F 01	Abrupt	43	2%	550 → 850
F 02	Abrupt	25	4%	310 → 650
F 03	Abrupt	30	3%	250 → 520
F 04	Random	34	0 → 2%	700 → 950
F 05	Random	44	0 → 3%	150 → 400
F 06	Random	08	0 → 4%	450 → 750
F 07	Additive (lin)	22	0 → 4%	610 → 950
F 08	Additive (lin)	16	0 → 2%	050 → 350
F 09	Additive (log)	02	0 → 3%	200 → 500
F 10	Intermittent	12	3%	670 → 700
		26	−3%	710 → 730
		06	2%	745 → 770
		24	−2.5%	780 → 800

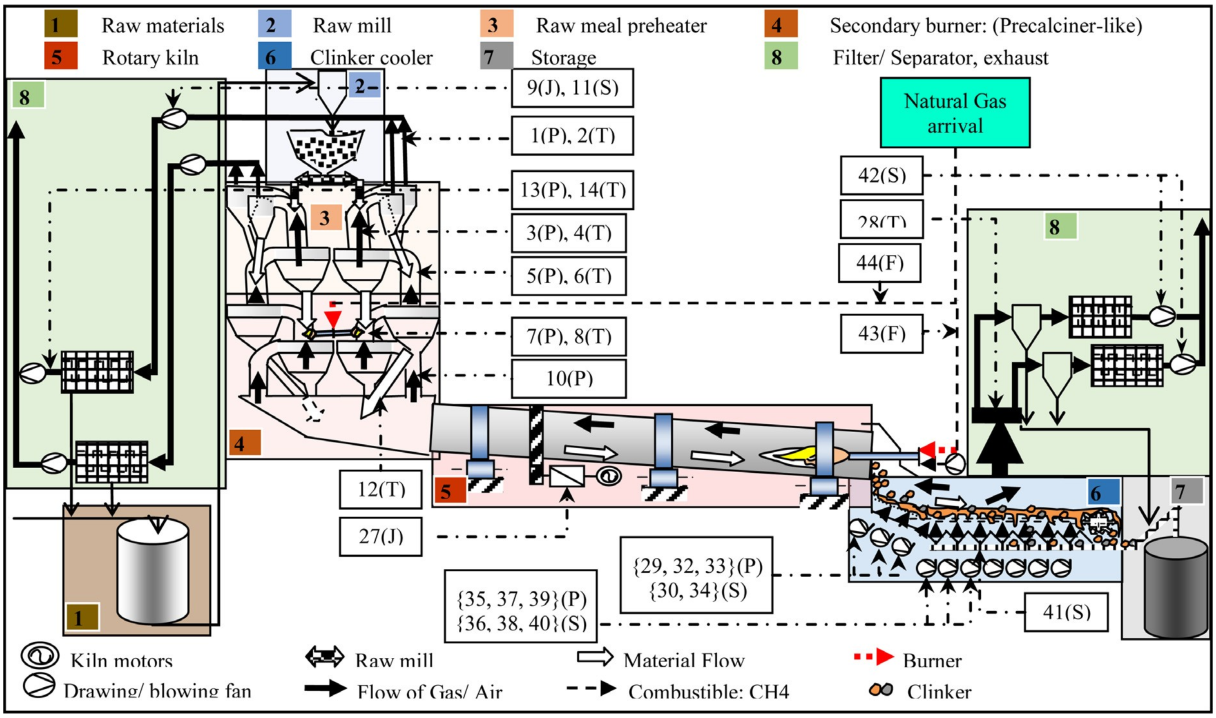

Figure 24: Schematic diagram of the CP rotary kiln.

4.3.2 Application Using KPCA

The real-world industry data set obtained from the cement plant is used now to test the proposed approaches to see how they perform in real-world scenarios.

Algorithm 3 is executed using the training data set which has 768 samples and 44 variables to build the monitoring model and algorithm 4 is used for the online monitoring, again a specified number of PCs is selected using CPV for each index to ensure the best performance possible of the model. For the T^2 index the number of PCs is 34, for the Q index it is 31, and for the combined index φ it is 12. The T^2 index has the largest number of PCs because this index is directly related to the principal component subspace.

Table 13 presents the result obtained using the conventional KPCA algorithm for different simulated faults and the real process fault (RPF), the conventional KPCA performs very well and it has successfully detected all faults with decent monitoring metrics. The fault, F 07, has high MDR and DTD values because it is a drift-wise type of fault and these faults are known for their late detection, one should take into account that this is not the case for all drift-wise faults, the other drift-wise faults, F 08 & F 09, have been detected instantly. The real process fault is also considered a drift-wise fault, KPCA has successfully detected this fault with a slightly high FAR value, this FAR value is acceptable and does not affect the monitoring performance.

Figure 25 show the detection process of the real process fault in the cement plant. it can be seen that the fault was detected successfully for different monitoring indices.

Table 13: KPCA Monitoring Results for the Cement Plant.

Indices	T^2			Q			φ		
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	0.92	0.00	0	0.92	0.00	0	0.17	0.00	0
F 02	1.29	0.00	0	1.04	0.00	0	0.26	0.00	0
F 03	1.14	0.00	0	0.98	0.00	0	0.24	0.00	0
F 04	1.20	3.19	0	0.96	2.79	0	0.24	3.98	0
F 05	1.04	15.54	0	0.72	23.11	0	0.24	18.73	1
F 06	0.92	8.64	0	0.92	7.64	0	0.17	12.62	0
F 07	0.95	14.37	47	0.95	14.66	33	0.17	21.99	57
F 08	0.42	6.64	0	0.25	14.95	0	0.08	7.97	3
F 09	1.17	0.00	0	1.00	0.00	0	0.25	0.00	0
F 10	0.86	0.00	0	0.79	0.00	0	0.14	0.00	0
RPF	27.14	0.54	0	10.71	1.20	1	16.67	0.96	1

4.3.3 Application Using Correlation Dimension RKPCA

To apply the Correlation Dimension RKPCA, it is necessary to check if this system is a chaotic system or not, this can be done by computing the largest Lyapunov Exponent of the training data set which is for this system is equal to 0.047 and since it is a positive value then this system is considered as chaotic and the Correlation Dimension RKPCA can be applied.

The correlation dimension of the cement plant is 10.98 and its ceiling is 11, by applying algorithm 5 the reduced matrix obtained should have only 11 samples. Figure 26 illustrates how the Correlation Dimension is computed using the $log\,(C_I)\ \ vs\ log\,(d)$ plot, the slope of the red line shown in the same figure is the value of CD.

The resulting reduced matrix from algorithm 5 is then used to build the monitoring model using algorithm 3 and then algorithm 4 to monitor the system. For the number of PCs selected for each index are 6 for T^2, 8 for Q, and 8 for φ. these number of PCs were selected for the best performance possible of the model. Table 14 contains the monitoring results obtained using this algorithm. The Correlation Dimension RKPCA has failed to detect the majority of faults for the T^2 index. For the Q index, the proposed approach in this part has a good monitoring performance except for the drift-wise type of faults but it has detected the real process fault better than the conventional KPCA technique which is good for such a small data set. The combined index, φ, is calculated as mentioned in 4.2.3 where the value of η for this system is 1 so the monitoring performance of φ is the same as the performance of the Q index.

The Correlation Dimension RKPCA can be implemented as a cyclic script in the monitoring software because it reduces the size of the data by a large amount so that it does not slow the execution performances of the monitoring main tasks and it can detect some faults with a successful rate as the RPF fault in the table 14.

Correlation Dimension RKPCA has failed to detect the real process fault for the T^2

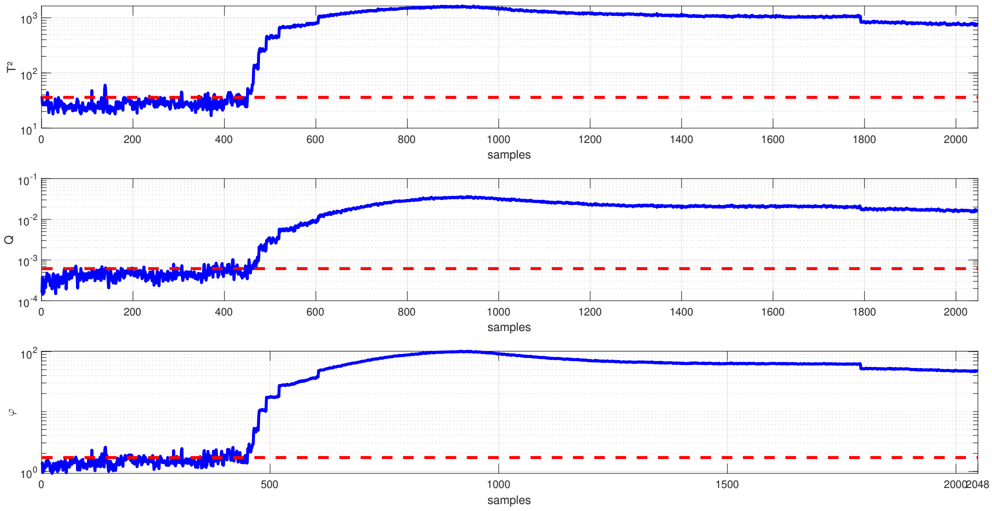

Figure 25: KPCA Monitoring for RPF.

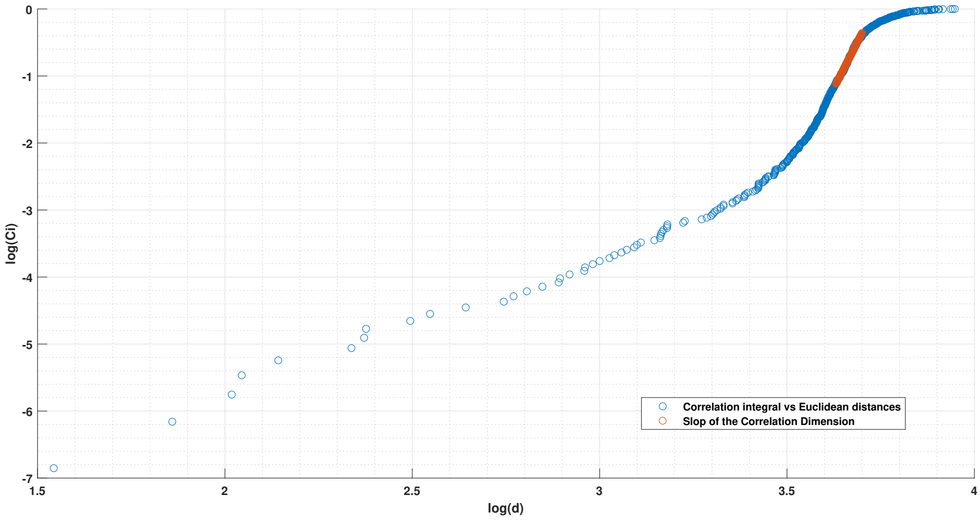

Figure 26: Estimation of CD (CP).

Table 14: Correlation Dimension RKPCA monitoring results

Indices	T^2			Q			φ		
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	0	100	NA	0.58	4.32	0	0.58	4.32	0
F 02	0	100	NA	0.78	16.13	0	0.78	16.13	0
F 03	0	0	0	0.65	0.00	0	0.65	0.00	0
F 04	0	44.62	0	0.48	3.19	0	0.48	3.19	0
F 05	0	100	NA	0.64	19.52	0	0.64	19.52	0
F 06	0	100	NA	0.67	8.64	0	0.67	8.64	0
F 07	0	100	NA	0.52	100	NA	0.52	100	NA
F 08	0	100	NA	0.58	92.69	7	0.58	92.69	7
F 09	0	100	NA	0.67	100	NA	0.67	100	NA
F 10	0	100	NA	0.57	26.26	0	0.57	26.26	0
RPF	0	100	NA	1.90	1.98	4	1.90	1.98	4

index but it was successful for the other two indices as shown in figure 27.

4.3.4 Application Using Variogram-based RKPCA

The Variogram-based RKPCA is directly applied to the system's data set without checking for certain specifications within the data set. Figure 7 is the Variogram of the cement plant data set, the red line is the sill of the variogram, and the green and black dashes are $c \pm \omega$. The selected lags h have variogram values within those two dashes.
For the cement plant data set, algorithm 6 is applied. If the selected values of ω are under 1×10^{-4} then there is no lag chosen and hence the reduced matrix is empty. If the values of ω are in the range of $[1.0 \times 10^{-4},\ 6.0 \times 10^{-4}[$ then only one lag is selected which is equal to 617 and this lag helps to create a reduced matrix with only 302 samples. When the ω values lies within the range $[6.0 \times 10^{-4},\ 1.1 \times 10^{-3}[$, these values leads to set of selected lags with the smallest one of 495 which then creates a reduced matrix of 546 samples. For $1.1 \times 10^{-3} \leq \omega < 1.4 \times 10^{-3}$, the smallest selected lag is 494 which helps to produce a reduced matrix of 548 samples. if ω is greater or equal 1.4×10^{-3} then the smallest selected lag is 141 which creates a full size reduced matrix. So, by applying algorithm 6 a set of reduced matrices is produced and the chosen one is selected due to the monitoring performances. Equation (6) is used to evaluate the performance of different matrices as presented in the following table 15. As can be seen the different resulting matrices from algorithm 6 have different monitoring performance.

Table 15: Monitoring Performance of Different Reduced Matrices Using Variogram

			T^2		Q		φ	
ω	h_m	Size	PC	J_{T^2}	PC	J_Q	PC	J_φ
$1 \times 10^{-4} \leq \omega < 6 \times 10^{-4}$	617	302	93	0.28	11	0.23	25	0.28
$6 \times 10^{-4} \leq \omega < 1.1 \times 10^{-3}$	495	546	40	**0.18**	34	**0.20**	31	**0.17**
$1.1 \times 10^{-3} \leq \omega < 1.4 \times 10^{-3}$	494	548	42	**0.18**	19	0.22	19	0.21

The minimum selected lag, h, is responsible for choosing the appropriate samples for

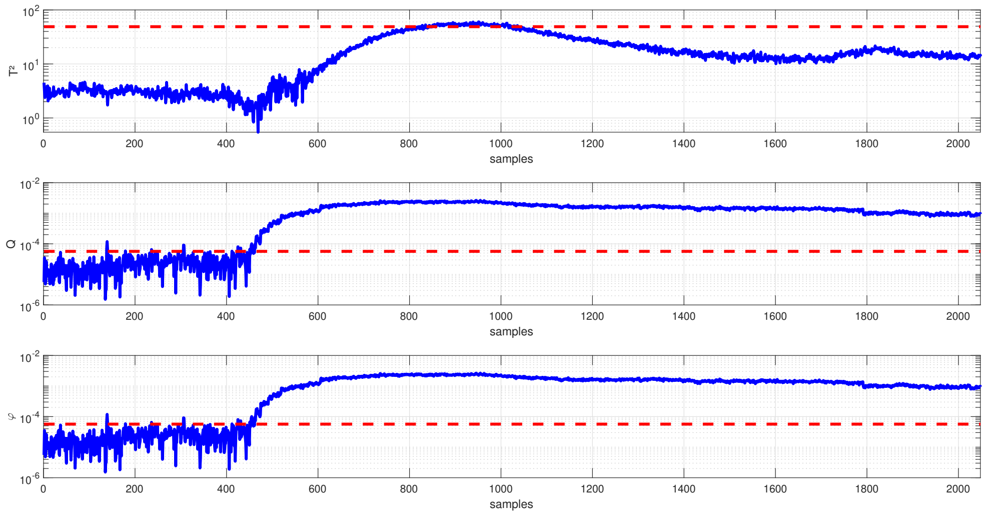

Figure 27: Correlation Dimension RKPCA Monitoring for RPF.

the reduced matrix. For the T^2 index, both matrices with 546 and 548 samples have the best monitoring performances with value of $J_{T^2} = 0.18$ using 40 PC and for the other two indices the first matrix with 546 samples has the best monitoring performance with $J_Q = 0.20$ with 34 PC and $J_\varphi = 0.17$ with 31 PC, hence it is the one selected for the monitoring model. Table 16 shows the monitoring metrics for this model.

Table 16: Variogram-based RKPCA Monitoring Performance

Indices		T^2			Q			φ	
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	0.50	0.00	0	0.75	0.00	0	0.33	0.00	0
F 02	0.69	0.00	0	1.47	0.00	0	0.52	0.00	0
F 03	0.65	0.00	0	1.38	0.00	0	0.49	0.00	0
F 04	0.64	3.19	0	0.48	4.78	0	0.48	2.79	0
F 05	0.56	24.70	1	1.28	25.10	1	0.48	23.90	1
F 06	0.50	7.97	0	1.33	7.97	0	0.33	7.64	0
F 07	0.52	17.01	54	0.43	9.97	33	0.35	13.78	33
F 08	0.25	6.64	0	1.08	14.62	0	0.33	12.96	0
F 09	0.67	0.00	0	1.42	0.00	0	0.30	0.00	0
F 10	0.43	0.00	0	1.07	0.00	0	0.29	0.00	0
RPF	18.10	0.78	0	17.86	1.68	3	7.62	1.50	1

For the T^2 index, the proposed approach has successfully detected all faults including the real process fault, unfortunately, it has a slightly high MDR value for the fifth and seventh simulated faults and high DTD value for the seventh fault this high detection delay value is due to the nature of fault which is drift-wise and hence it affects the missed alarm rate value. For the Q index, the false alarm rate of all faults is acceptable and does not affect the performance of the monitoring model, the missed detection rate is generally acceptable except for the fifth simulated fault as for the detection delay the seventh simulated fault has a somewhat high value. The combined index, φ, has a better general performance than the other two indices, the FAR values are all acceptable, especially for the real process fault, the MDR values are better than the values of the other two indices except for the seventh and eighth faults. The DTD values are similar to the ones of the Q index. In general, the Variogram-based RKPCA has successfully reduced the size of the training data set and maintained a decent overall monitoring performance which is the purpose of this study.

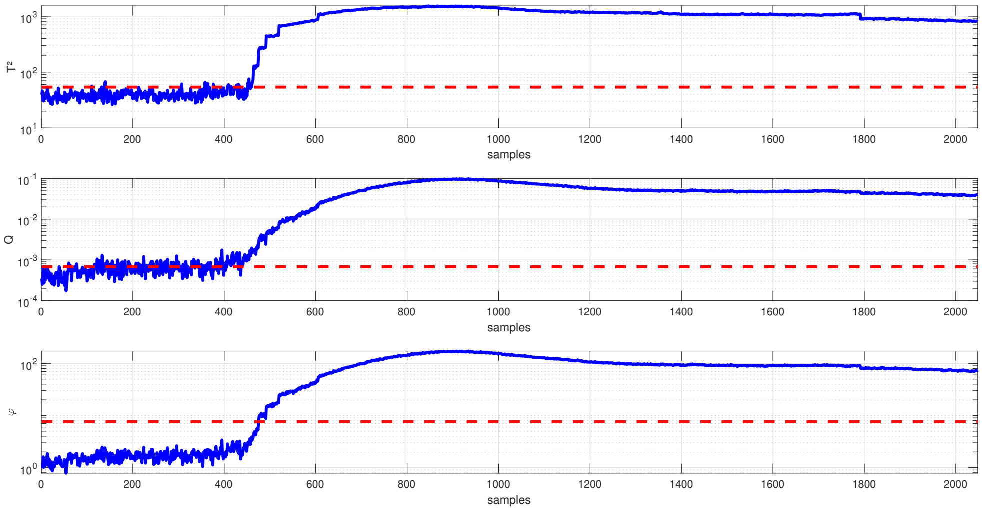

Figure 28: Variogram-based RKPCA Monitoring RPF.

The Variogram-based RKPCA has a good monitoring result for the real process fault despite some high FAR values, figure 28 shows these performances for different indices.

4.3.5 Application Using Histogram-based RKPCA

Algorithm 7 is used to reduce the number of samples in the training data set collected from the Cement Plant. Depending on N_B, the Histogram-based RKPCA produces a set of reduced matrices and again the cost function, J_s, is used to evaluate the performance of each one of them. The resulting reduced matrix is related to both ε and N_B, for some reduced data it can have the same minimum appearance frequency but they are not the same matrices.

Table 17 presents results obtained using Histogram-based RKPCA on the cement plant data. From this table, it can be seen that a different matrix has outstanding performance for various indices. For the T^2 index, matrices resulting from $N_B = \{20 \ \& \ 12\}$ have the best overall performance regarding this index and it can be noticed that all matrices have a close number of retained PCs. Matrix obtained using $N_B = 09$ leads the monitoring performances for the Q index, Unlike the first monitoring index the range of retained PCs is larger. finally, matrices from $N_B = \{20, \ 16, \ 15, \ \& \ 11\}$ have the best monitoring

Table 17: Monitoring Performances of Different Reduced Matrices on Cement Plant.

		$T2$		Q		φ	
N_B	ϵ	PC	J_{T^2}	PC	J_Q	PC	J_φ
20	02	36	**0.17**	20	0.20	30	**0.18**
18	03	40	0.20	25	0.21	25	0.20
16	04	44	0.18	10	0.20	22	**0.18**
15	04	35	0.18	47	0.17	19	**0.18**
12	03	37	**0.17**	27	0.19	25	0.19
11	05	37	0.19	11	0.19	26	**0.18**
10	06	38	0.18	43	0.18	20	0.19
09	07	35	0.20	30	**0.15**	28	0.19
08	08	32	0.23	14	0.21	24	0.19
07	09	32	0.23	14	0.21	24	0.19
06	13	31	0.25	29	0.26	22	0.21

performances regarding the φ index and the number of retained PCs is close to each other. So if the monitoring system is built upon one index then one of these matrices is selected, but if this system depends on all indices then the matrix with the best all-around performance is selected. The comparison is now based on the cost function from (7) which is the mean of all J_s. Matrix is obtained by $N_B = 15$ have the best monitoring performance based on J with value of 0.18. This matrix is then used to compare its performance with other algorithms.

Table 18: Histogram-based RKPCA Monitoring Performance for Cement Plant

Indices		T^2			Q			φ	
Metrics	FAR	MDR	DTD	FAR	MDR	DTD	FAR	MDR	DTD
F 01	0.25	0.00	0	1.25	0.00	0	0.67	0.00	0
F 02	0.26	0.00	0	1.38	0.00	0	1.12	0.00	0
F 03	0.24	0.00	0	1.30	0.00	0	1.06	0.00	0
F 04	0.24	3.19	0	1.44	3.98	0	0.88	3.19	0
F 05	0.24	20.72	1	1.28	40.24	1	1.04	17.13	1
F 06	0.25	11.96	0	1.33	12.62	0	0.92	11.30	0
F 07	0.26	21.99	66	1.29	21.41	1	0.69	19.35	56
F 08	0.08	20.60	0	0.83	14.62	6	0.67	13.62	0
F 09	0.25	0.00	0	1.33	0.00	0	1.08	0.00	0
F 10	0.21	0.00	0	1.21	0.00	0	0.79	0.00	0
RPF	6.67	1.02	0	11.19	0.96	0	17.14	0.72	0

From table 18, it can be noticed that the Histogram-based RKPCA has successfully detected all faults for different monitoring indices. From the same table, the Histogram-based RKPCA has some difficulty detecting F 05 and F 06 faults which are random because it has relatively high MDR values for those two faults, also Fault F 07 has both MDR and DTD values. F 08 has a high MDR value, one should consider that F 07 and F 08 are a stepwise type of faults. For the rest of the faults, the proposed algorithm performs as well as anticipated.

Figure 29 shows the histogram of the first PC of the training data set without reduction and figure 30 shows the histogram of the reduced data set. From these figures, it can be noticed that both data have the same distribution and the values of the appearance frequency are divided by four which is the same value as the minimum appearance frequency.

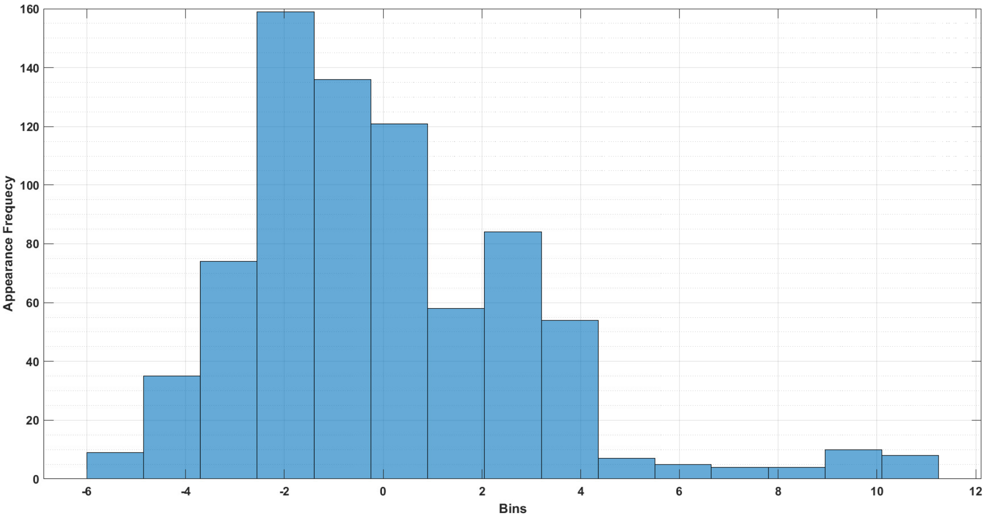

Figure 29: Histogram of the 1^{st} PC Score of Original Data (CP).

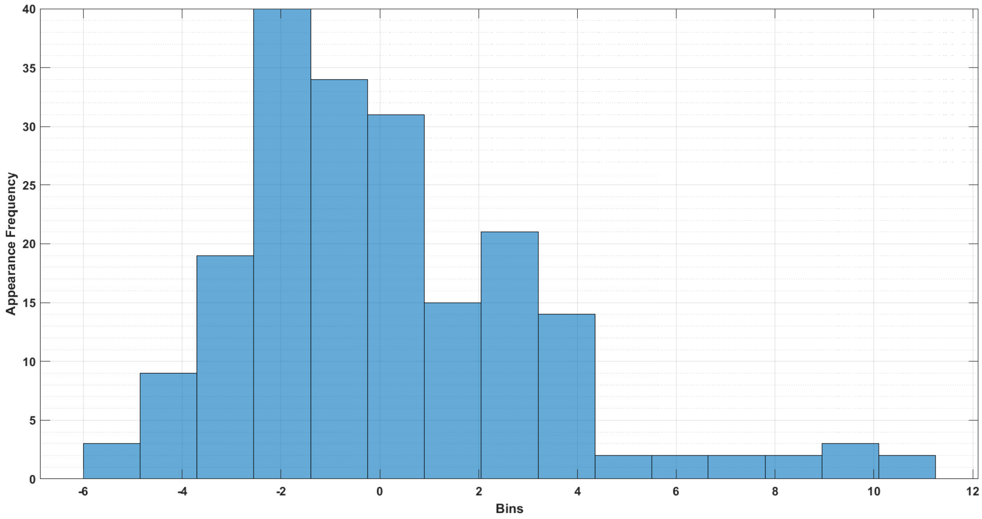

Figure 30: Histogram of the 1^{st} PC Score of Original Data (CP).

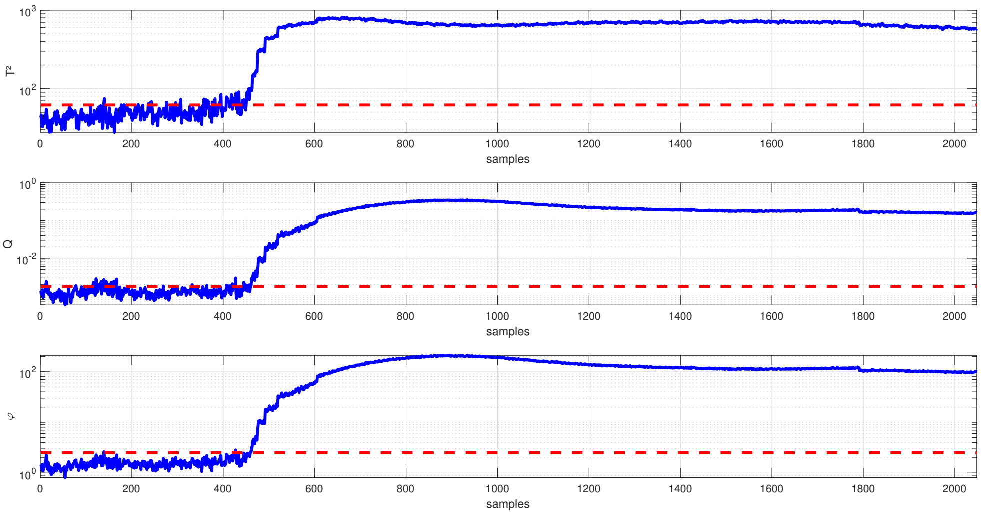

Figure 31: Histogram-based RKPCA Monitoring for RPF.

Figure 31 illustrates the monitoring performances of the Histogram-based RKPCA using different indices to detect the real process fault, this algorithm has successfully detected this fault without any flop.

4.3.6 Results and Discussion for the Cement Plant

In evaluating the performance of monitoring systems constructed using a single monitoring index, a comparative analysis was conducted between the proposed algorithms and the conventional KPCA. The cost function, denoted as J_s, was utilized to determine which matrix provided the best performance. This comparison encompasses all matrices generated by the reduction methods as well as the matrix from the KPCA algorithm. The results are based on tables 13, 14, 16, and 18

For the T^2 index, the Histogram-based RKPCA approach utilises reduced matrices obtained using $N_B = 20$ or $N_B = 12$, delivers the best performance with a cost function value of $J_{T^2} = 0.17$. This indicates superior monitoring capabilities compared to other methods. Following closely are the Variogram-based RKPCA matrices, which consist of 546 samples, and the conventional KPCA, both achieving a monitoring performance value of $J_{T^2} = 0.18$. These results are competitive but slightly less effective than the Histogram-based RKPCA. The Correlation Dimension RKPCA ranks last in this comparison, with a significantly higher cost function value of $J_{T^2} = 1.67$, reflecting much poorer performance.

For the Q index, the Histogram-based RKPCA again outperforms all other methods with a cost function value of $J_Q = 0.15$, achieved using a reduced matrix obtained by $N_B = 09$. This represents the best monitoring performance for this index. The conventional KPCA algorithm follows with a performance value of $J_Q = 0.17$, and the Variogram-based RKPCA is next with $J_Q = 0.20$. The Correlation Dimension RKPCA is at the bottom with a $J_Q = 0.60$, indicating the least effective performance among the methods tested.

For the combined index φ, The Variogram-based RKPCA and KPCA methods both achieve the best performance with a cost function value of $J_\varphi = 0.17$. These methods provide optimal monitoring results for the φ index. The Histogram-based RKPCA follows closely with a $J_\varphi = 0.18$, showing competitive performance but not quite as strong as the top methods. The Correlation Dimension RKPCA once again has the poorest performance with a $J_\varphi = 0.60$, reflecting its consistent under-performance across the indices. Overall, for monitoring models based on individual indices, both the Variogram-based RKPCA and Histogram-based RKPCA approaches have improved upon the monitoring performance of the conventional KPCA algorithm. In contrast, the Correlation Dimension RKPCA has not achieved the desired results and falls short of expectations in enhancing monitoring performance.

This section focuses on a comparative analysis of monitoring performance among three proposed algorithms and several existing methods. The comparison is conducted using a model that incorporates all indices for monitoring. Additionally, the algorithms are evaluated based on various other criteria, including the storage space required, the computation time needed, and the homogeneity between the original and reduced datasets.

Table 19 presents the results of different algorithms evaluated across various monitoring indices. The comparison reveals the performance metrics for each algorithm, providing insights into their effectiveness and efficiency. Based on the T^2 index, The KPCA algorithm, Variogram-based RKPCA, and Histogram-based RKPCA exhibit the lowest values of J_{T^2}, indicating the best performance for this index. Among these, the Histogram-based RKPCA stands out with the smallest size of the reduced matrix, which suggests it offers superior performance in monitoring compared to the others. Following these top performers, the Euclidean Distance RKPCA is next, with performance slightly trailing behind. The k-means RKPCA, Reduced Rank KPCA, and Correlation Dimension RKPCA follow in sequence, with the Correlation Dimension RKPCA performing the least effectively.

The KPCA algorithm, k-means RKPCA, and Histogram-based RKPCA achieve the best monitoring performance. Notably, the k-means RKPCA appears to have the most efficient reduced dataset size for the Q index. The Euclidean Distance RKPCA performs slightly less effectively. The remaining algorithms are ranked as follows: Variogram-based RKPCA, Reduced Rank RKPCA, and Correlation Dimension RKPCA, with the latter showing the least favorable results.

For the φ index, the k-means RKPCA algorithm demonstrates exceptional performance, outperforming the conventional KPCA for the first time. This represents a significant achievement, highlighting the effectiveness of the k-means RKPCA in handling combined index data. The KPCA and Variogram-based RKPCA algorithms follow closely, both showing strong performance. The Histogram-based RKPCA is next in line, offering solid but slightly less effective monitoring compared to the top methods. The final three algorithms—Euclidean Distance RKPCA, Reduced Rank RKPCA, and Correlation Dimension RKPCA—show poor performance. Particularly, the Correlation Dimension RKPCA has consistently underperformed across all indices. This poor performance is likely due to the significantly reduced number of observations retained, which impacts its ability to effectively monitor the data.

Table 19: CP Cost Function J_s values for Different Algorithms.

		T^2		Q		φ	
Method	Size	PCs	J_s	PCs	J_s	PCs	J_s
KPCA	768	39	**0.18**	31	**0.17**	30	0.17
Reduced Rank RKPCA [40]	613	25	0.51	10	0.27	10	0.26
Variogram-based RKPCA [62]	546	40	**0.18**	34	0.20	31	0.17
Correlation Dimension RKPCA [63]	11	4	1.67	8	0.60	8	0.60
Euclidean Distance RKPCA [19]	131	38	0.35	22	0.18	18	0.19
k-means RKPCA [41]	131	18	0.44	08	**0.17**	18	**0.11**
Histogram-based RKPCA [64]	198	35	**0.18**	47	**0.17**	19	0.18

If a user opts to use a monitoring model that incorporates all three indices simultaneously, the overall performance is assessed using the cost function J, which represents the average of the cost functions for each individual index. The analysis reveals that the conventional KPCA algorithm achieves the best overall monitoring performance, with a J value of 0.17. This is closely followed by both the Variogram-based RKPCA and Histogram-based RKPCA approaches, each attaining a J value of 0.18. In third place, the k-means RKPCA and Euclidean Distance RKPCA both exhibit a J value of 0.24. The Reduced Rank RKPCA comes next with a higher J value of 0.35, indicating a decline in performance compared to the top methods. Finally, the Correlation Dimension RKPCA has the poorest overall performance, with a J value of 0.96.

Notably, in terms of monitoring performance relative to the size of the training dataset, the Histogram-based RKPCA stands out as the leader among the methods evaluated. This suggests that it does not only perform well across the indices but also manages to effectively handle the dataset size, enhancing its overall monitoring capabilities.

In the homogeneity comparison, the Variogram-based RKPCA reduced dataset exhibits only one non-homogeneous variable, specifically (x_{15}), when compared to the original dataset. This suggests that the Variogram-based RKPCA method retains a high degree of similarity to the original data. The Reduced Rank RKPCA, Euclidean Distance RKPCA, and Histogram-based RKPCA approaches each have two non-homogeneous variables, namely x_{15}, x_{44}. This indicates a slightly higher level of deviation from the original dataset, but still maintains a relatively good level of homogeneity. In contrast, the k-means RKPCA approach reveals four non-homogeneous variables, indicating a more significant divergence from the original dataset compared to the previously mentioned methods. The Correlation Dimension RKPCA stands out with a notably higher number of non-homogeneous variables, totaling seventeen. This large number suggests a substantial discrepancy from the original data, reflecting a significant loss of homogeneity.

Table 20 presents the results of the homogeneity test, underscoring the effectiveness of the Variogram-based RKPCA and Histogram-based RKPCA in maintaining the homogeneity of the reduced dataset. Conversely, the Correlation Dimension RKPCA shows a considerable shortfall in preserving data homogeneity, highlighting its limitations in effective data reduction.

Table 20: CP non-homogeneous variables for different algorithms.

Method	Non-Homogeneous variables
Reduced Rank RKPCA [40]	$2 \sim [x15, x44]$
Variogram-based RKPCA [62]	$1\sim [x15]$
Correlation Dimension RKPCA [63]	$17 \sim [x1, x8, x10, x14, x17, x18, x20,$ $x29, x34, \ldots, x41, x44]$
Euclidean Distance RKPCA [19]	$2 \sim [x15, x44]$
k-means RKPCA [41]	$4 \sim [x1, x10, x15, x44]$
Histogram-based RKPCA [64]	$2 \sim [x15, x44]$

The RKPCA approach provides a notable solution for enhancing the KPCA algorithm by reducing both the required storage space and the execution time for one sample during the online phase. These improvements are crucial for optimizing the efficiency of the KPCA model. Table 21 offers detailed information on these two criteria for comparison purposes. This table highlights how the RKPCA approach impacts the storage requirements and processing time, thereby offering insights into its efficiency benefits. To better understand the relationship between the number of samples and the execution time, the following equations are provided. These equations illustrate how the execution time scales with changes in the number of samples, thereby quantifying the impact of the number of samples on computational performance.

$$T^2 \rightarrow \begin{cases} E\left(n\right) = 7.71 \times 10^{-12} n^3 - 2.454 \times 10^{-9} n^2 + 8.566 \times 10^{-7} n + 3.945 \times 10^{-6}, & 0 \leq n \leq 361 \\ E\left(n\right) = 1.115 \times 10^{-12} n^3 + 3.352 \times 10^{-9} n^2 + 2.193 \times 10^{-6} n + 0.0005, & n > 361 \end{cases}$$

$$Q \rightarrow \begin{cases} E\left(n\right) = 1.129 \times 10^{-11} n^3 - 3.409 \times 10^{-9} n^2 + 9.4 \times 10^{-7} n + 4.291 \times 10^{-7}, & 0 \leq n \leq 361 \\ E\left(n\right) = 1.617 \times 10^{-11} n^3 - 2.058 \times 10^{-8} n^2 + 1.491 \times 10^{-5} n + 0.002537, & n > 361 \end{cases}$$

$$\varphi \rightarrow \begin{cases} E\left(n\right) = 1.375 \times 10^{-11} n^3 - 3.744 \times 10^{-9} n^2 + 9.777 \times 10^{-7} n + 3.77 \times 10^{-6}, & 0 \leq n \leq 361 \\ E\left(n\right) = 2.145 \times 10^{-11} n^3 - 2.981 \times 10^{-8} n^2 + 2.012 \times 10^{-5} n - 0.003475, & n > 361 \end{cases}$$

As observed in the previous subsection, there is a direct relationship between the required storage space and the number of observations in the training dataset, as well as the execution time. Table 21 provides insights into this relationship. From the table, it is evident that the Correlation Dimension RKPCA requires the smallest storage space due to its significantly lower number of observations. This is followed by the Euclidean Distance RKPCA and k-means RKPCA, which also demonstrate relatively low storage requirements. The Histogram-based RKPCA comes next in terms of required storage space.

In terms of execution time, the ranking of the algorithms aligns with their storage space requirements. Therefore, the Correlation Dimension RKPCA, with its minimal storage needs, also has the shortest execution time, followed by the Euclidean Distance RKPCA and k-means RKPCA. The Histogram-based RKPCA ranks next, with Variogram-based RKPCA, Reduced Rank RKPCA, and conventional KPCA requiring more storage and longer execution times.

These results highlight that managing a very large number of samples can pose challenges for monitoring systems. It is important to note that these performance metrics are also influenced by the hardware used to implement the monitoring system. Hence, hardware specifications play a crucial role in determining the overall efficiency of these algorithms.

Table 21: CP execution time and required storage space

			T^2	Q	φ
Method	Size	Storage Space (ko)	Execution Time (ms)	Execution Time (ms)	Execution Time (ms)
KPCA	768	8704	3.90	4.20	4.30
Reduced Rank RKPCA [40]	613	5298	2.40	2.60	2.70
Variogram-based RKPCA [62]	546	4532	2.00	2.20	2.20
Correlation Dimension RKPCA [63]	**11**	4	2.72×10^{-2}	2.43×10^{-2}	2.43×10^{-2}
Euclidean Distance RKPCA [19]	131	257	0.12	0.12	0.13
k-means RKPCA [41]	131	259	0.12	0.12	0.13
Histogram-based RKPCA [64]	198	594	0.21	0.25	0.26

4.4 Conclusion

For the TEP benchmark evaluation, the proposed approaches clearly demonstrated success in reducing both the required storage space and execution time compared to the conventional KPCA algorithm. However, regarding monitoring performance, the Correlation Dimension RKPCA did not meet expectations, failing to deliver satisfactory results. In contrast, the Variogram-based RKPCA exhibited the best monitoring performance among the proposed methods, though it did not achieve a significant reduction in dataset size. The Histogram-based RKPCA, however, effectively retained approximately half of the original samples while still providing respectable monitoring performance. These observations highlight the Histogram-based RKPCA as the most effective approach for TEP, successfully balancing the trade-off between dataset size reduction and strong monitoring performance. When applied to the CP data, the Variogram-based RKPCA and Histogram-based RKPCA performed as expected, often leading in monitoring performance across various scenarios. Both methods consistently demonstrated strong capabilities, maintaining acceptable performance even when not achieving the absolute best results. Conversely, the Correlation Dimension RKPCA underperformed in nearly all aspects of monitoring when compared to the other proposed algorithms. Nevertheless, it offers a distinct advantage in terms of required storage space and execution time due to the significantly smaller number of observations it retains, showcasing efficiency in these specific areas despite its inferior monitoring performance. The comprehensive evaluation in this chapter confirms that the proposed algorithms performed as anticipated, outperforming some existing algorithms in various cases. They effectively optimized execution time and minimized storage requirements without sacrificing overall performance. Furthermore, with the notable exception of the Correlation Dimension RKPCA, the proposed methods successfully retained homogeneity with the original data. These findings collectively affirm the significant potential of the proposed methods for robust fault detection.

General Conclusion

This dissertation introduced three novel RKPCA algorithms to address key limitations of traditional Kernel PCA, particularly those related to the size of the training dataset. Each algorithm aimed to mitigate these issues while maintaining or even improving monitoring performance. The Histogram-based RKPCA leverages class intervals for data reduction, robustly preserving the original distribution without prior assumptions; this method demonstrates a superior balance between significant data reduction and impressive monitoring performance. The Variogram-based RKPCA uses spatial continuity to eliminate correlated samples from the training set, also without requiring data assumptions; this approach consistently shows robust monitoring performance, making it optimal when detection accuracy is paramount, even if the degree of data reduction is less critical. Finally, the Correlation Dimension RKPCA, designed specifically for chaotic systems, excels at achieving substantial data reduction, making it a highly viable option for integration into monitoring software as a cyclic script, particularly when storage efficiency and processing speed are paramount.

The proposed algorithms were rigorously tested on the Tennessee Eastman Process and the Ain El Kebira Cement Plant to comprehensively evaluate their efficacy against existing methods. The Variogram-based RKPCA and Histogram-based RKPCA successfully achieved the study's core objectives by effectively addressing KPCA's drawbacks while maintaining strong performance across various metrics. The Correlation Dimension RKPCA, however, partially fulfilled these objectives due to its inconsistent monitoring performance. Compared to published works, the Variogram-based RKPCA and Histogram-based RKPCA exhibited good performance for both processes, aligning well with methods like k-means RKPCA and Euclidean Distance RKPCA. The Correlation Dimension RKPCA notably falls short in comparison, especially concerning its monitoring performance. The three proposed algorithms contributed to notably reduced execution times and lower storage requirements than KPCA.

Despite their individual strengths, these algorithms do come with certain limitations. The Correlation Dimension method, while excellent for chaotic systems, is only effective for specific types of faults, and its monitoring performance isn't consistently stable. Both the Variogram-based and Histogram-based methods can generate multiple reduced matrices, which might add complexity in certain applications. Furthermore, the Variogram method's ability to reduce data isn't always guaranteed, as its effectiveness depends significantly on the minimum lag selected.

Future research aims to build upon this thesis's contributions in RKPCA for fault detection. Although the proposed methods have demonstrated significant advances in computational efficiency and performance, there remains considerable room for further enhancement in their robustness, broadening their applicability, and addressing specific challenges, such as the limitations observed with the Correlation Dimension RKPCA. These future endeavours will focus on refining algorithmic capabilities, exploring new validation scenarios, and delving into theoretical extensions to further advance the state-of-the-art in intelligent fault detection systems.

List of Publications

I. Mohammed Tahar Habib Kaib et al. “RKPCA-based approach for fault detection in large scale systems using variogram method”. In: *Chemometrics and Intelligent Laboratory Systems* 225 (2022), p. 104558. ISSN: 0169-7439. DOI: `https://doi.org/10.1016/j.chemolab.2022.104558`. URL: `https://www.sciencedirect.com/science/article/pii/S0169743922000697`

II. Mohammed Tahar Habib Kaib et al. “Improving kernel PCA-based algorithm for fault detection in nonlinear industrial process through fractal dimension”. In: *Process Safety and Environmental Protection* 179 (2023), pp. 525–536. ISSN: 0957-5820. DOI: `https://doi.org/10.1016/j.psep.2023.09.010`. URL: `https://www.sciencedirect.com/science/article/pii/S0957582023008212`

III. Mohammed Tahar Habib Kaib et al. “Improvement of Kernel Principal Component Analysis-Based Approach for Nonlinear Process Monitoring by Data Set Size Reduction Using Class Interval”. In: *IEEE Access* 12 (2024), pp. 11470–11480. DOI: `10.1109/ACCESS.2024.3354926`

IV. Mohammed Tahar Habib Kaib et al. “Data size reduction approach for nonlinear process monitoring refinement using Kernel PCA technique”. In: *Expert Systems with Applications* 274 (2025), p. 126975. ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2025.126975`. URL: `https://www.sciencedirect.com/science/article/pii/S0957417425005974`

V. Mohammed Tahar Habib Kaib et al. “Kernel Principal Component Analysis Improvement based on Data-Reduction via Class Interval”. In: *IFAC-PapersOnLine* 58.4 (2024). 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2024, pp. 390–395. ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2024.07.249`. URL: `https://www.sciencedirect.com/science/article/pii/S2405896324003331`

References

[1] Anam Abid, Muhammad Tahir Khan, and Javaid Iqbal. "A review on fault detection and diagnosis techniques: basics and beyond". In: *Artificial Intelligence Review* 54.5 (2021), pp. 3639–3664.

[2] Janos Gertler. *Fault Detection and Diagnosis*. 2015.

[3] Leo H Chiang, Evan L Russell, and Richard D Braatz. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[4] Venkat Venkatasubramanian et al. "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods". In: *Computers & chemical engineering* 27.3 (2003), pp. 293–311.

[5] Paul M. Frank. "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results". In: *Automatica* 26.3 (1990), pp. 459–474. ISSN: 0005-1098. DOI: `https://doi.org/10.1016/0005-1098(90)90018-D`. URL: `https://www.sciencedirect.com/science/article/pii/000510989090018D`.

[6] R.J. Patton and J. Chen. "A Review of Parity Space Approaches to Fault Diagnosis". In: *IFAC Proceedings Volumes* 24.6 (1991). IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'91), Baden-Baden, Germany, 10-13 September 1991, pp. 65–81. ISSN: 1474-6670. DOI: `https://doi.org/10.1016/S1474-6670(17)51124-6`. URL: `https://www.sciencedirect.com/science/article/pii/S1474667017511246`.

[7] Bruce K Walker and Kuang-yang Huang. "FDI by extended Kalman filter parameter estimation for the industrial actuator benchmark". In: *IFAC Proceedings Volumes* 27.5 (1994), pp. 481–487.

[8] J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1996.

[9] Venkat Venkatasubramanian, Raghunathan Rengaswamy, and Surya N Kavuri. "A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies". In: *Computers & chemical engineering* 27.3 (2003), pp. 313–326.

[10] Daniel Dvorak and Benjamin Kuipers. "Model-Based Monitoring of Dynamic Systems." In: *IJCAI*. 1989, pp. 1238–1243.

[11] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[12] Venkat Venkatasubramanian et al. "A review of process fault detection and diagnosis: Part III: Process history based methods". In: *Computers & chemical engineering* 27.3 (2003), pp. 327–346.

[13] Brijen R Bakshi and George Stephanopoulos. "Reasoning about trends in process engineering". In: *Computers & Chemical Engineering* 18.4 (1994), pp. 213–241.

[14] John F MacGregor and Theodora Kourti. "Statistical process control of multivariate processes". In: *Control engineering practice* 3.3 (1995), pp. 403–414.

[15] Zongli Ge, Chunhua Yang, and Xiliang Chen. “A review of data-driven process monitoring and fault diagnosis”. In: *IEEE Transactions on Industrial Informatics* 13.3 (2017), pp. 1044–1055.

[16] Ying-Shu Huang and Peng-Keng Li. “Fault diagnosis using neural networks: A survey”. In: *International Journal of Computer Applications in Technology* 18.1 (2003), pp. 24–34.

[17] JC Benneyan, RC Lloyd, and PE Plsek. “Statistical process control as a tool for research and healthcare improvement”. In: *BMJ Quality & Safety* 12.6 (2003), pp. 458–464.

[18] Mustapha Ammiche. “Online thresholding techniques for process”. PhD. University M’Hamed Bougara of Boumerdes, 2018.

[19] F Bencheikh et al. “New reduced kernel PCA for fault detection and diagnosis in cement rotary kiln”. In: *Chemometrics and Intelligent Laboratory Systems* 204 (2020), p. 104091.

[20] Ian T Jolliffe and Jorge Cadima. “Principal component analysis: a review and recent developments”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[21] Mark A Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE journal* 37.2 (1991), pp. 233–243.

[22] Shufeng Tan and Michael L Mayrovouniotis. “Reducing data dimensionality through optimizing neural network inputs”. In: *AIChE Journal* 41.6 (1995), pp. 1471–1480.

[23] Dong Dong and Thomas J McAvoy. “Nonlinear principal component analysis—based on principal curves and neural networks”. In: *Computers & Chemical Engineering* 20.1 (1996), pp. 65–78.

[24] HG Hiden et al. “G., A. Montague,“Non-linear Principal Components Analysis Using Genetic Programming””. In: *Genetic algorithms in engineering systems: Innovations and Applications*. Vol. 446, pp. 302–307.

[25] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* 10.5 (1998), pp. 1299–1319.

[26] H. Abdi and L. J. Williams. “Principal component analysis”. In: *WIREs Computational Statistics* 2 (4 2010), pp. 433–459. DOI: 10.1002/wics.101.

[27] Sabrina Bennai. “Reduced Kernel PCA based Approach for Fault Detection in Complex Systems”. Masters’ degree Thesis. Boumerdes: University M’Hamed BOUGARA, 2019.

[28] Abdelhalim Louifi et al. “Sensor Fault Detection in Uncertain Large-Scale Systems Using Interval-Valued PCA Technique”. In: *IEEE Sensors Journal* 25.2 (2025), pp. 3119–3125. DOI: 10.1109/JSEN.2024.3507876.

[29] Lokesh Gour et al. “Characterization of rice (Oryza sativa L.) genotypes using principal component analysis including scree plot & rotated component matrix”. In: *International Journal of Chemical Studies* 5.4 (2017), pp. 975–83.

[30] J Brown. “Choosing the right number of components or factors in PCA and EFA”. In: *JALT Testing & Evaluation SIG Newsletter* 13.2 (2009).

[31] Julie Josse and François Husson. “Selecting the number of components in principal component analysis using cross-validation approximations”. In: *Computational Statistics & Data Analysis* 56.6 (2012), pp. 1869–1879. ISSN: 0167-9473. DOI: `https://doi.org/10.1016/j.csda.2011.11.012`. URL: `https://www.sciencedirect.com/science/article/pii/S0167947311004099`.

[32] Jinxin Wang et al. “A multivariate statistics-based approach for detecting diesel engine faults with weak signatures”. In: *Energies* 13.4 (2020), p. 873.

[33] H Henry Yue and S Joe Qin. “Reconstruction-based fault identification using a combined index”. In: *Industrial & engineering chemistry research* 40.20 (2001), pp. 4403–4414.

[34] Mark A Kramer. “Autoassociative neural networks”. In: *Computers & chemical engineering* 16.4 (1992), pp. 313–328.

[35] Majdi Mansouri et al. “Kernel PCA-based GLRT for nonlinear fault detection of chemical processes”. In: *Journal of Loss Prevention in the Process Industries* 40 (2016), pp. 334–347.

[36] M-F Harkat et al. “Machine learning-based reduced kernel PCA model for nonlinear chemical process monitoring”. In: *Journal of Control, Automation and Electrical Systems* 31.5 (2020), pp. 1196–1209.

[37] Karl Ezra Pilario et al. “A review of kernel methods for feature extraction in nonlinear process monitoring”. In: *Processes* 8.1 (2019), p. 24.

[38] Karen Kazor et al. “Comparison of linear and nonlinear dimension reduction techniques for automated process monitoring of a decentralized wastewater treatment facility”. In: *Stochastic environmental research and risk assessment* 30.5 (2016), pp. 1527–1544.

[39] Feng Zhao et al. “Two-Phase Incremental Kernel PCA for Learning Massive or Online Datasets”. In: *Complexity* 2019.1 (2019), p. 5937274.

[40] Hajer Lahdhiri et al. “Nonlinear process monitoring based on new reduced Rank-KPCA method”. In: *Stochastic Environmental Research and Risk Assessment* 32.6 (2018), pp. 1833–1848.

[41] Carlos Felipe and Alcala Perez. “FAULT DIAGNOSIS WITH RECONSTRUCTION-BASED CONTRIBUTIONS FOR STATISTICAL PROCESS MONITORING”. PhD thesis. Viterbi School of Engineering, 2011.

[42] Khadija Attouri et al. “Improved fault detection based on kernel PCA for monitoring industrial applications”. In: *Journal of Process Control* 133 (2024), p. 103143. ISSN: 0959-1524. DOI: `https://doi.org/10.1016/j.jprocont.2023.103143`. URL: `https://www.sciencedirect.com/science/article/pii/S0959152423002317`.

[43] Okba Taouali et al. “New fault detection method based on reduced kernel principal component analysis (RKPCA)”. In: *The International Journal of Advanced Manufacturing Technology* 85.5 (2016), pp. 1547–1552.

[44] Peiling Cui, Junhong Li, and Guizeng Wang. "Improved kernel principal component analysis for fault detection". In: *Expert Systems with Applications* 34.2 (2008), pp. 1210–1219. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2006.12.010. URL: https://www.sciencedirect.com/science/article/pii/S095741740600409X.

[45] G. Baudat and F. Anouar. "Kernel-based methods and function approximation". In: *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222).* Vol. 2. 2001, 1244–1249 vol.2. DOI: 10.1109/IJCNN.2001.939539.

[46] Adeel Aslam Bhutta. "Chaos Theory & Fractals". In: *Applied Signal Processing* (1999).

[47] James Theiler. "Estimating fractal dimension". In: *JOSA A* 7.6 (1990), pp. 1055–1073.

[48] Steven H Strogatz. *Nonlinear dynamics and chaos.* en. Boca Raton: Chapman and Hall/CRC, Jan. 2024.

[49] Septima Poinsette Clark. "Estimating the fractal dimension of chaotic time series". In: *Lincoln Laboratory Journal* 3.1 (1990).

[50] Wahyu Caesarendra et al. "An application of nonlinear feature extraction-A case study for low speed slewing bearing condition monitoring and prognosis". In: *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.* IEEE. 2013, pp. 1713–1718.

[51] Geoff Bohling. "Introduction to geostatistics and variogram analysis". In: *Kansas geological survey* 1 (2005), pp. 1–20.

[52] Randal Barnes. "Variogram tutorial". In: *Golden, CO: Golden. Software. Available online at http://www. goldensoftware. com/. variogramTutorial. pdf* (2003).

[53] Catherine Calder and Noel A Cressie. "Kriging and variogram models". In: (2009).

[54] Abdelmalek Kouadri, Mohanad Amokrane Aitouche, and Mimoun Zelmat. "Variogram-based fault diagnosis in an interconnected tank system". In: *ISA transactions* 51.3 (2012), pp. 471–476.

[55] J. Chen. *Histogram.* Investopedia. Aug. 2021. URL: https://www.investopedia.com/terms/h/histogram.asp.

[56] Alan Agresti. *Categorical data analysis.* John Wiley & Sons, 2013.

[57] Andy Field. *Discovering statistics using IBM SPSS statistics.* Sage publications limited, 2024.

[58] Frederick J Gravetter and Larry B Wallnau. "Statistics for the behavioral sciences 10th". In: *Statistic for The Behavioral Science* (2017).

[59] Max Wornowizki and Roland Fried. "Two-sample homogeneity tests based on divergence measures". In: *Computational Statistics* 31.1 (2016), pp. 291–313.

[60] James J Downs and Ernest F Vogel. "A plant-wide industrial process control problem". In: *Computers & chemical engineering* 17.3 (1993), pp. 245–255.

[61] Georg A Gottwald and Ian Melbourne. "A new test for chaos in deterministic systems". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 460.2042 (2004), pp. 603–611.

[62] Mohammed Tahar Habib Kaib et al. "RKPCA-based approach for fault detection in large scale systems using variogram method". In: *Chemometrics and Intelligent Laboratory Systems* 225 (2022), p. 104558. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2022.104558. URL: https://www.sciencedirect.com/science/article/pii/S0169743922000697.

[63] Mohammed Tahar Habib Kaib et al. "Improving kernel PCA-based algorithm for fault detection in nonlinear industrial process through fractal dimension". In: *Process Safety and Environmental Protection* 179 (2023), pp. 525–536. ISSN: 0957-5820. DOI: https://doi.org/10.1016/j.psep.2023.09.010. URL: https://www.sciencedirect.com/science/article/pii/S0957582023008212.

[64] Mohammed Tahar Habib Kaib et al. "Improvement of Kernel Principal Component Analysis-Based Approach for Nonlinear Process Monitoring by Data Set Size Reduction Using Class Interval". In: *IEEE Access* 12 (2024), pp. 11470–11480. DOI: 10.1109/ACCESS.2024.3354926.

[65] Mohammed Tahar Habib Kaib et al. "Data size reduction approach for nonlinear process monitoring refinement using Kernel PCA technique". In: *Expert Systems with Applications* 274 (2025), p. 126975. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2025.126975. URL: https://www.sciencedirect.com/science/article/pii/S0957417425005974.

[66] Mohammed Tahar Habib Kaib et al. "Kernel Principal Component Analysis Improvement based on Data-Reduction via Class Interval". In: *IFAC-PapersOnLine* 58.4 (2024). 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2024, pp. 390–395. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2024.07.249. URL: https://www.sciencedirect.com/science/article/pii/S2405896324003331.

Appendix A: Mercers' Theorem

Mercers' Theorem plays a crucial role in KPCA because the mapping function can be replaced by a kernel function $\kappa(x,\ y)$.

Mercers' Theorem: $\kappa(x,\ y)$ is a continuous, symmetric, and positive semi-definite kernel function defined on a compact domain $\iota \times \iota$, there exists a set of orthonormal eigen function $\Phi_i(a)$ and corresponding non-negative eigenvalues λ_i such that a kernel function can be expressed as

$$\kappa(x,\ y) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(x) \Phi_i(y)$$

where the series converges absolutely and uniformly. So, as long as κ is a valid Mercer kernel it corresponds to an inner product in some high-dimensional feature space.