Data-Driven Process Monitoring, Fault Detection And Fault Diagnosis

by

Bahador Rashidi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Control Systems

Department of Electrical and Computer Engineering

University of Alberta

© Bahador Rashidi, 2020

Abstract

As the process control industry and production lines become highly complex and significantly invested with high-dimensional variables, process health monitoring attracts more attention from the domain experts and process operators. Since data is ubiquitous nowadays thanks to the advanced computer and communication technology, data-driven approaches are frequently used to ensure the safety and process quality performance. The proposed research in this thesis is mainly focused on two aspects: fault detection in non-stationary processes and root-cause fault diagnosis using causality analysis.

In Chapter 2, fault detection is investigated from an unsupervised perspective for processes with non-stationary time-series measurements, especially those subject to time-varying mean changes. To this end, a moving-mean principal component analysis (MM-PCA) approach is proposed, in which the mean values of process measurements are updated using a moving-mean algorithm based on the upper bound of expected range of variations. The proposed MM-PCA does not require a heavy online calculation in comparison with the existing adaptive solutions and it can successfully compartmentalize the faults from healthy variations. Applying the concept of MM-PCA, three monitoring feature indices are proposed to monitor the statistical behavior of the process measurements. Moreover, an overall health index is suggested based on the proposed features using kernel density estimators (KDEs) which is considered as a process condition indicator.

In Chapter 3, quality output-related fault detection based on non-stationary process measurement is studied. A cascaded modeling framework based on partial least-squares (PLS) approach is introduced, which entails a complete orthogonal projection of the process variables onto quality output-related and quality output-unrelated subspaces. The principal manifold is defined to represent the underlying auto-regressive model of the time-series, and such a relationship remains unchanged during the normal time-varying operations. Consequently, proper quality output-related and unrelated indices are derived.

In a majority of multivariate processes, the propagating nature of abnormalities makes root-cause fault diagnosis a challenging task. As the second main focus of this research, we endeavor to develop a root-cause fault diagnosis framework based on causality analysis using transfer entropy (TE). With this aim, in Chapter 4, a novel data-driven strategy is proposed for real-time root-cause fault diagnosis in (non-)linear processes by estimating the causal strength between measured process variables and variations of a residual signal (e.g. square prediction error derived by PCA or kernel PCA) using normalized transfer entropy (NTE). A novel approach for TE estimation, i.e. the so-called symbolic dynamic-based normalized transfer entropy (SDNTE) is proposed, which has achieved a faster computation speed and less complexity than the conventional KDE method. For this purpose, a new definition of joint xD-Markov machine is given to capture dynamic interactions between two time-series. The concept of SDNTE is built upon principles of time-series symbolization, joint xD-Markov machine, and joint-Shannon entropies. Not only the SDNTE has less calculation complexity in comparison to KDE approach, but also the proposed general framework in this chapter can effectively identify the source of the process fault among certain potential candidates. The proposed root-cause fault diagnosis framework is applied to the Tennessee Eastman Process (TEP) benchmark and its computational advantages are clearly demonstrated.

Finally in Chapter 5, a complete autonomous framework is proposed for conducting root-cause fault diagnosis which requires a minimum *a priori* process knowledge and intervention of a human operator. Upon the presence of a fault, potential process variables are identified using a contribution score algorithm and SDNTE is used for generating the directed graph which presents the causal inference among the candidates. Then, Direct transfer entropy (DTE) is utilized to prune the indirect and spurious edges. To this aim, the application of symbolic dynamic filtering (SDF) is extended to the propose symbolic dynamic normalized direct transfer entropy (SDNDTE). Accordingly, concepts of immediate and source intermediate variables are defined and autonomous algorithms are developed to efficiently find them using the initial causal graph. In the end, a depth-first search (DFS)-based algorithm is developed and deployed on the pruned graph to locate the root-cause variable(s).

Preface

This thesis is an original work by Bahador Rashidi which has been done under the supervision of Prof. Qing Zhao. A part of research work in this thesis is supported by the industry partner(s) and the results are filed as US patents. The other research results are published as scholar articles.

- The results of Chapter 2 were filed as a pending US patent: B. Rashidi, M. S. Krishnaswamy, Q. Zhao, *Feature extraction and fault detection in a non-stationary process through unsupervised machine learning*, Pending US Patent, sponsored by Honeywell Process Solution, 16/014, 542, 2019. This industrial application was an original research work done by B. Rashidi who was responsible for developing the underlying approach and implementation as well as the composition of the manuscript. M. S. Krishnaswamy was the lead contact engineer of the industry partner, who was involved in the application filing and data gathering. Q. Zhao was the supervisory author and involved in concept formation and patent document revisions.

- The results of Chapter 3 have been published in the article: B. Rashidi, Q. Zhao, *Quality-Related Fault Detection for Processes with Time-Varying Measurements*, 2019 18th European Control Conference (ECC), Napoli, Italy, July 2019, pp. 3873-3879. This was an original research work done by B. Rashidi who was responsible for developing the underlying approach and implementation as well as the composition of the manuscript. Q. Zhao was the supervisory author and involved in concept formation and manuscript revisions.

- The results of Chapter 4 have been published in the article: B. Rashidi, D. S. Singh, Q. Zhao, *Data-driven root-cause fault diagnosis for multivariate non-linear processes*, Control Engineering Practice, Elsevier, Volume 70, Pages 134-147, Nov 2017. This research and its industrial application were original works done by B. Rashidi who was responsible for developing the main underlying approaches, simulations, and plant data analysis as well as the composition of the manuscript. The PDF D. S. Singh contributed to the symbolic dynamic filtering part of the article as an advisory author. Q. Zhao was the supervisory author and involved in concept formation and manuscript revisions.

- A general industrial application for data-driven process health monitoring was proposed, in which each chapter of this thesis can be used as a component. This work was filed as a pending US patent: M. S. Krishnaswamy, B. Rashidi, Q. Zhao, *Autonomous predictive real-time monitoring of faults in process and equipments*, Pending US Patent, sponsored by Honeywell Process Solution, file number 16/012, 542, 2019. This industrial application was an original research/industrial work done by B. Rashidi who was responsible for developing the underlying approach and implementation as well as the composition of the manuscript. M. S. Krishnaswamy was the lead contact engineer of the industry partner, who was involved in the application filing, data gathering and implementation advices. Q. Zhao was the supervisory author and involved in concept formation and manuscript revisions.

To my beloved parents, and my brother Behrad
for their endless love and support

Acknowledgements

Firstly, I would like to express my sincere appreciation to Prof. *Qing Zhao* for her great support and supervision during this research project. This research and dissertation would not have been possible without her continuous guidance. To my Ph.D. committee members, I am extremely grateful for their assistance and suggestions throughout the study and examination process.

Last but not least, my deepest gratitude goes to my brother *Behrad* and my future business partner *Mohamed Abed* for their unconditional support during my graduate studies.

Bahador Rashidi
Edmonton, Alberta
Canada

Contents

List of Tables

List of Figures

Notation

$\mathbb{R}$	The sets of real number
$\mathbb{R}^n$	The set of real n-dimensional vector
$\mathbb{R}^{n \times m}$	The set of real $n \times m$ matrices
$x \in X$	x is an element of set X
$X \subseteq Y$	X is a subset or equal set of Y
A^{T}	Transpose of matrix A
A^{-1}	Inverse of matrix A
$A^{\dagger}$	Pseudo inverse of matrix A
$\psi^{\perp}$	Perpendicular subspace of ψ
$a == b$	If a is equal to b
$s \leftarrow x_s$	Append x_s into s
$\mathcal{N}(\mu, \sigma)$	The normal distribution with mean μ and variance σ
$H(.\vert.)$	Conditional entropy
$H(.\vert.,.)$	Conditional joint-entropy
$SDH(.\vert.)$	Symbolic dynamic conditional entropy
$SDH(.\vert.,.)$	Symbolic dynamic conditional joint-entropy
$P(.\vert.)$	Conditional probability distribution function
$P(.,.,.)$	Joint probability distribution function
$\lvert\Sigma^x\rvert$	Cardinality of symbols for time-series x
$\sigma_i \in \mathit{\Sigma}^x$	i^{th} symbol in the symbolic sequence
$\lvert Q^x\rvert$	Cardinality of states of symbolic sequence of time-series x
$q_i^x \in Q^x$	i^{th} state in the symbolic state sequence
$q_i^{x_a x_b} \in Q^{x_a x_b}$	symbolic state of joint sequence of $\{x_a x_b\}$
π_{ij}^{xx}	Probability state matrix for symbolic state sequence q^x
$\tilde{\pi}^{xx}$	Morph emission matrix
$\mathcal{A}^{x_a x_b \rightarrow x_b}$	represents a $8 - \mathit{tuple}$ *xD-Markov* machine
$PA^{\bar{G}}(x_i)$	Parents of node x_i in graph $\bar{G}$

Abbreviations

PCA	Principal component analysis
MM-PCA	Moving-mean PCA
KPCA	Kernel PCA
FAR	False alram rate
SVD	Singular value decomposition
UCL	Upper control limit
PDF	Probability density function
KDE	Kernel density estimators
SPE	Square prediction error
PLS	Partial Least-squares
KPLS	Kernel PLS
TE	Transfer Entropy
DTE	Direct transfer entropy
SDH	Symbolic dynamic *Shannon* entropy
SDNTE	Symbolic dynamic normalized transfer entropy
SDNDTE	Symbolic dynamic normalized direct transfer entropy
IIV	Immediate intermediate variable
SIV	Source intermediate variable
CSTR	Continuous stirred tank reactor
TEP	Tennessee Eastman process
IFMOC	Identifiable functional model classes
PFSA	Probabilistic finite state automata

Chapter 1

Introduction and Motivation

1.1 Background and Research Scope

Process health monitoring techniques have been widely applied in industrial processes to effectively enhance safety and reliability as well as reduce maintenance costs by detecting anomalies in time. This line of work also plays a prominent role in the design and implementation of a reliable and cost-efficient control system. Initiated in the early 1970s, model-based fault detection and diagnosis, also known as quantitative approaches have been significantly developed since then. A few years later, qualitative data-driven methods were introduced for the same purpose, specifically when there is no clear knowledge about the system base-line model. Many qualitative and quantitative methods have been proposed and well summarized in surveys [3], [4] and [5]. Application of model-based techniques [4] [6] [7] [8] may lead to difficulties in implementation due to high complexity of the industrial processes and lack of information about their model structures. On the other hand, data-driven methods [9] [10] [11] [12] are simpler and easier to implement which can be performed effectively without the need for *a priori* knowledge of the process model. Hence, the focus of this thesis is on the application of data-driven approaches for process monitoring and fault diagnosis.

In Fig. 1.1, the general road map for a typical process monitoring investigated in this thesis is depicted. Given a process, the first step after conducting necessary pre-processing is to detect the presence of a fault by monitoring the measured time-series. This step alone has been discussed in numerous studies for different case scenarios and applications with the topic of fault detection. The statistical nature of the measurement (e.g. stationarity) plays an important role in the selection of the right approach and most of conventional fault detection methods impose stationary assumptions which are not the case in the majority of industrial processes. To this end, two chapters of this thesis are centered around addressing the fault detection problem for the process with non-stationary measurements.

As can be seen in Fig. 1.1, the second step after detecting a fault is to identify the variables that are affected by it. Some methodologies [13] [14] [2] can flag faulty variables, but due to the propagating nature of the fault in most cases, the flagged variables may not be the true source of the fault. Although process operators find it useful to know the faulty variables in a

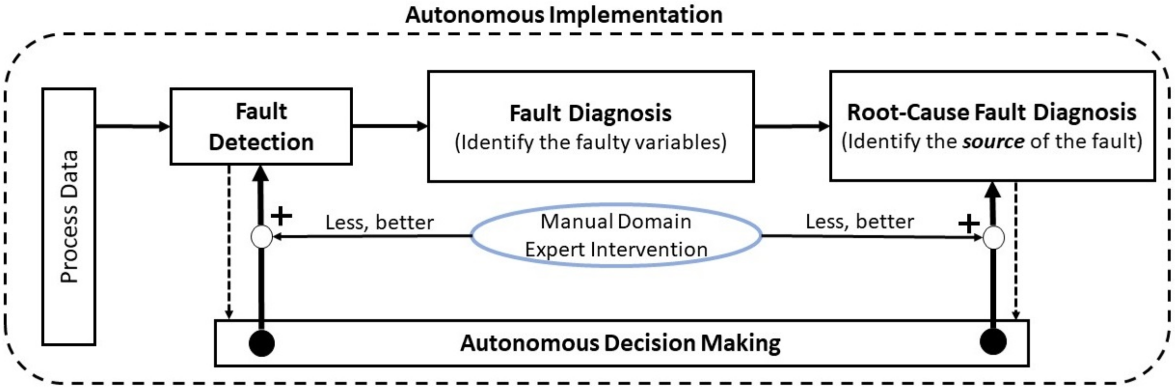

Autonomous Implementation
Process Data
Fault Detection
Fault Diagnosis
(Identify the faulty variables)
Root-Cause Fault Diagnosis
(Identify the *source* of the fault)
+
Less, better
Manual Domain Expert Intervention
Less, better
+
Autonomous Decision Making

Figure 1.1: The general block diagram for the thesis research scope.

high-dimensional process in order to have a better understanding of the process health status, this information is not sufficient to identify the faulty components due to the propagating nature of the faults.

One of the timely demands of process operators is to gain knowledge about the root-cause of the detected fault. This information is mainly required to conduct preventive actions to the system to avoid reaching critical conditions or maintain the quality of key performance indicators (KPIs) by conducting proactive maintenance actions. According to the type of the process, e.g. (non-)linear and/or (non-)stationary, various methodologies may be chosen for conducting causality analysis and further identifying the root-cause of the detected fault. In this thesis, two chapters are dedicated to this topic, based on transfer entropy analysis in a new symbolic dynamic formulation for conducting causality analysis and root-cause diagnosis of the detected fault. For the application of causality analysis based root-cause fault diagnosis, there are various tuning parameters, and manual selection/design needs to be done by the domain experts who are assumed to know the topology of the system under study. One of the examples of how *a priori* knowledge and intervention of the domain expert is necessary can be found in [15]. Heavy tuning and dependence on operator knowledge may hinder the application of such a technique. For this reason, Chapter 5 proposes an autonomous framework for the suggested root-cause fault diagnosis which requires less human intervention and has less computational complexity. Reducing the domain experts' interactions in implementation of process health monitoring methodologies has received remarkable attention in both literature and industry and it is also one of the main topics in machine learning [16] [17].

1.2 Literature Survey

- **Fault Detection in Non-Stationary Processes:**

An industrial system can be classified with respect to different system properties such as linear/non-linear, and time-invariant/time-varying system [18]. Similarly, based on the properties of time-series, one can treat process measurement that follows a distribution with constant mean and variance, as the stationary process, or the one with varying mean/variance as the non-stationary process. Non-stationary measurements may be the result of a stochastic system in which the base-line parameters are subjected to changes or random variations. Consequently, this results in mean and variance changes in the time-series probability distributions. In addition, some other cases that lead to non-stationarity are manipulated inputs (e.g. intentional changes or close-loop compensation effect) and/or other internal process actions such as material degradation (e.g. catalyst degradation in CSTR process [19]), corrosion and valve/nozzle plugging (e.g. residue plugging in high-speed centrifuges [20]), etc. In [21], co-integration is considered as an assumption for non-stationary time-series to provide one solution. Other research work that adopts a co-integrated structure for non-stationary time-series to create health monitoring indices can be found in [22] [18].

Among all available data-driven methods, PCA [23], canonical variate analysis (CVA) [24], independent component analysis (ICA) [25] and partial least-squares (PLS) [26] have been frequently used for fault detection. From the implementation perspective, each one of the aforementioned methods has pros and cons in comparison with other counterparts [9]. Principal component analysis (PCA) [27] and its various modified versions [23] [28] [29] were utilized for different types of processes. For the multivariate non-linear cases, kernel PCA is widely used [30] [31] [32] [33]. By utilizing *kernel* trick [34], KPCA firstly maps the process variables with non-linear relations onto a high-dimensional feature space and then applies the standard PCA for generating statistical indices to monitor the process. Ordinary PCA [23] and partial least-squares (PLS) [35] were initially proposed to monitor linear stationary processes such that the relationship between measurements follow a static and linear pattern. The other underlying assumption behind these two methods is that the measurement time-series follow *Gaussian* distribution. To handle dynamic relationship between process time-series, the *Augmentation* approach was proposed in [23] to take into account the auto-correlation and cross-correlation, which led to proposing dynamic PCA (DPCA) to identify the base-line model of the chemical process.

PCA is a straightforward yet powerful method for fault detection and has been implemented in many process monitoring products, and its modified and improved versions are of great interest. On the other hand, fault detection in non-stationary processes is still an ongoing challenge, particularly for the high-dimensional industrial processes with non-stationary mean variations. Alongside with the ordinary PCA sequels, two viable schemes are adaptive and recursive PCA [36] [37]. These methods suggest updating the mean, covariance matrix and number of principal components in a block-wise manner. Hence, it requires to conduct singular value decomposition (SVD) upon arrival of a new block of the test data which can be computationally involved and requires an enormous amount of attention from the operators for parameter tuning. The other limitation of adaptive PCA is that it attempts to update the base-line model with any changes in the process unless the change is relatively abrupt and violates a tuning threshold. In other words, whenever an abnormal block of data is observed, a decision index is calculated and it is compared with a corresponding threshold. If the decision index crosses the threshold, the base-line model is getting updated, otherwise, it concurs that a fault exists in the process. Furthermore, relatively complex parameter tuning steps that are required for updating the threshold violate the simplicity of the original PCA approach and hinder its industrial implementations.

- **Output-Related Fault Detection in Non-Stationary Processes:**

To build a linear relationship between process measurements and the process quality output, PLS [38] is commonly used. The core idea of ordinary PLS for output prediction is to extract a number of latent variables from highly correlated measurements based on the covariance between process variables and quality outputs [35]. For monitoring processes with the static relationship between measured variables and quality outputs, there have been a variety of modified versions of standard PLS. In [39], a recursive PLS scheme is proposed which updates the latent model with the most recent process measurements. Yin *et al.* studied some other modifications on PLS algorithm for output monitoring purposes [40]. In [41], Qin and Zheng utilized concurrent projection in the structure of PLS and proposed the concurrent-PLS method as an efficient process monitoring tool. Moreover, Zhou *et al.* proposed total projection to latent structure (T-PLS) [42], in which a post-processing step is added to further decompose the scores and loading matrices of standard PLS. Although T-PLS extracts more information about the impact of detected fault on the quality outputs, it uses oblique projection which does not guarantee complete decomposition of quality-related information from the quality-unrelated counterpart. To address this problem, Yin *et al.* proposed improved PLS (IPLS) [43] algorithm which leverages complete orthogonal projection to totally decompose the quality output-related information from the process variables. For non-linear cases, similar to KPCA, kernel PLS (KPLS) [44] was proposed to remedy output-related fault detection problem. Majority of conventional fault detection methods have a relatively strong stationarity assumption. Moreover, applications of these PLS-based algorithms are usually limited to the processes, in which the relationship between process variables and quality outputs is static. One solution to this limitation is to construct augmented matrices from process variables in order to incorporate the inner dynamic interactions [23]. Based on this idea, Jiao *et al.* has proposed a dynamic improved PLS algorithm [1], which is claimed to be more efficient than its previously developed counterparts.

In general, the aforementioned techniques are able to monitor the quality of a stationary process in which the operating-point(s) or mean-value(s) of both process variables and quality outputs are time-invariant. In other words, if the mean of process measurement has a continuous variation, the typical monitoring statistics used in previous PLS-based methods may falsely flag the normal time-varying operating condition as an output-related malfunction. In [45] and [46], multi-mode PLS schemes are proposed to tackle quality-related fault detection for non-stationary multi-mode processes. In [47], a recursive T-PLS approach is proposed to update the loading matrices upon receiving the new data. In the updating mechanism proposed in this paper, singular value decomposition is conducted every time which adds computational complexity for real-time applications. On the other hand, certain research works is focused on the modification of the PLS algorithm without using a moving window or other real-time adaptation mechanism. The aim of such an approach is to directly tackle non-stationary variation inherent in certain processes such as non-isothermal continuous stirred tank reactor (CSTR) or acetylene hydrogenation process [48].

- **Application of Causality Analysis for Root-Cause Fault Diagnosis:**

In general, the underlying idea of PCA-based algorithms including kernel PCA is to find a correlation(s) among process variables. Therefore, certain common methods for fault diagnosis, such as contribution plot analysis [14] and accumulative rate contribution score [2] incorporated with PCA and kernel PCA, may suffer from smearing-out effects as a result of fault propagation [49] [14]. Consequently, these methods may not be able to locate the fault root-cause in an industrial process. This limitation is due to ignoring causal relationships among process variables, which are of great importance for identifying the fault source(s). According to the type of the process (e.g. (non-)linear and/or (non-)stationary), various methodologies may be utilized for conducting causality analysis and further identifying the root-cause of the detected fault. For causality analysis among stationary time-series, there already exist several methods including spectral envelope, adjacency matrix, Bayesian network inference, Granger causality (GC) and transfer entropy (TE). However, for the non-stationary counterpart, the application of the aforementioned methods might lead to an erroneous result, thus other alternatives such as dynamic time warping-based analysis [50] can be applied to identify the root-cause of the malfunction.

Among the existing methods, spectral envelope is presented as a causality analysis scheme in frequency domain [51]. A graph-based method so-called adjacency matrix [52] is another technique strictly dependent on the process model, which is not always available especially for complex industrial systems. Bayesian network (BN) inference [53], as a direct acyclic graph (DAG)-based method, is applicable to cases where less amount of historical data is available. BN method generally suffers from high computational complexity and may be used for risk assessment purposes in industrial processes. Granger causality (GC) is another common scheme that finds the causal relations among time-series utilizing the regression structure (e.g. ARX and AR). This method is easy to implement and has low computational complexity, but it is not generally applicable to the case when the time-series have non-linear relationships [54] [55].

Transfer entropy (TE) originally proposed in [56] is another conventional and viable tool for finding causality between two time-series. TE has been widely adopted in different industrial and neuroscience applications. In [57] [58], TE is applied to find cause and effect relationships (causal map) among process measurements in the multivariate industry process. Moreover, Le *et al* utilized TE to find the fault root-cause among the potential faulty candidates selected from all process variables by utilizing a reconstruction-based contribution method. There exist other modified versions of TE such as direct transfer entropy (DTE) [59] and transfer zero entropy (T0E) [60], which provide more explicit information under certain assumptions about the existing direct pathways between time-series.

The key point of the TE approach for causality analysis is that it relies on the distribution (i.e. joint probability density functions) of the process variables rather than their regression model, which is the case in Granger causality. Hence, TE can be applied for both linear and non-linear processes [59]. This advantage of the TE over GC is the main reason for it to be used for root-cause fault diagnosis in this thesis. However, as mentioned in [50] [57] [59], causality analysis using TE requires a burdensome computational effort and may not be applicable for real-time root-cause diagnosis. The reason behind this computational obstacle is that in almost all of the proposed TE-based approaches, joint probability density functions (PDFs) in the definition of TE are estimated by kernel density estimator [61], which has high computational order and requires significant amount of temporal data. Therefore, this computational complexity limits application of TE-based methods to off-line causality analysis in industrial processes. In order to address this limitation of the TE method for real-time root-cause fault diagnosis application, we propose a new and fast symbolic dynamic-based pathway for estimating transfer entropy between time-series, which has significantly lower computational order and requires less amount of temporal data. For completeness, symbolic dynamic filtering (SDF) method is introduced at first and its procedure is reviewed in section 4.4.2.

- **Root-Cause Fault Diagnosis Using TE and DTE:**

In the literature, there are certain research works on root-cause fault diagnosis using TE and DTE, in which some unresolved challenges exist. For instance, in [50] and [62], TE was utilized to generate a directed causal graph amongst candidate variables that are selected by reconstruction-based contribution index and modified canonical variable analysis (MCVA), respectively. *A priori* knowledge about the base-line models was utilized to locate the source of the faults. Furthermore, in the causal map indirect and spurious connections were not distinguished from a direct path for finding root-cause variables, which may lead to false diagnosis. In order to eliminate the indirect connections in the causal map, an approach is proposed in [63] by conducting a dedicated search algorithm on the *a priori* topology of the process which might not be available for any industrial cases. Moreover, Ma *et al* [64] recently has adopted DTE to prune the indirect connections in an initial causal map generated by TE. Due to the presence of the intermediate variables (IVs) in the definition of the DTE, the computation complexity of PDF estimation increases drastically and becomes even worse than the estimation of TE. Furthermore, this approach also requires *a priori* knowledge about the process to manually determine the potential indirect pathways and there is no algorithmic way to efficiently find the intermediate variables for calculating the DTE. One of the motivations that we pursue in this research is to develop a systematic way to find the right IVs in the implementation of DTE.

In summery, the application of TE and DTE in a systematic framework to identify the source variable(s) among potential candidates has been a point of interest. Although TE and DTE have been developed and utilized for root-cause fault diagnosis in [62], [64] and [15], the lack of automation in implementation and the level of domain expert intervention pose limitations which have not been fully addressed.

1.3 Objectives and Contributions

The general objective of this thesis is to propose data-driven solutions for the aforementioned process health monitoring problem statements specifically in fault detection of non-stationary systems and real-time root-cause fault diagnosis. For this purpose, two chapters are dedicated to studying fault detection in non-stationary processes that violate the fundamental assumptions in some state-of-the-art methods using PCA and PLS. Then in two other chapters, the root-cause fault diagnosis is investigated for both linear and non-linear processes with relatively low calculation complexity in comparison with the other existing solutions. Moreover, an autonomous framework is proposed to reduce the dependency of real-time implementation on the *a priori* process knowledge and the domain expert intervention.

- **Chapter 2**

The core objective of this chapter is to tackle anomaly detection in non-stationary industrial processes with unexpected manipulated set-point changes and uncertainties in the prior knowledge about the statistical nature of the measurements. In this chapter, the fault detection is investigated from an unsupervised perspective such that a moving-mean PCA approach is proposed, which utilizes the base-line loading matrices and a defined upper bound for the expected variation range to loosen the stationarity assumption. Hence, the mean values that are being used for normalizing the time-series are adaptively updated without any need for a real-time recalculation as in other existing solutions. Moreover, the first- and second-order error indices are defined to monitor a wide range of dynamic changes.

This chapter proposes a hybrid framework that integrates the application of principal component analysis (i.e. kernel trick transformation for non-linear cases) and an unsupervised probability-based anomaly detection method for (non-)linear processes with non-stationary measurements. The following lists main contributions of the proposed solution that has been submitted as a pending US patent[1].

[1] B. Rashidi, M. S. Krishnaswamy, Q. Zhao, On the application of unsupervised machine learning for fault detection in linear non-stationary industrial process, Pending US Patent, sponsored by Honeywell Company, file H0081201, 2017-2018.

1- This framework is proposed to tackle the problem of fault detection in non-stationary processes which has not been well addressed by other similar approaches. In other words, the proposed paradigm is a suitable anomaly detection tool for the majority of an industrial process which is composed of manipulated variables subjected to non-stationary changes. The proposed strategy does not require a heavy online updating calculation upon arrival of the new test data, hence, the computational cost of the proposed method is much less than existing schemes such as moving-window PCA [65] and adaptive PCA [36].

2- Novel feature indices are proposed to be extracted from a non-stationary time-series, which have clear physical interpretations for a better understanding of the users and monitor the higher-order changing behaviour of process measurements.

3- Along with these features, a novel health index is introduced with some favourable properties to distinguish normal non-stationary changes from different types of process faults.

- **Chapter 3**

In this chapter, a novel least-squares-based scheme is proposed for quality-related fault detection of the dynamic linear processes, in which the measured variables and quality-outputs may be subjected to time-variant changes. The proposed method is built upon a cascade modeling framework including complete orthogonal projection of the process variables onto output(quality)-related and output(quality)-unrelated subspaces and finding the principal manifolds for the projected components which represent their underlying auto-regressive moving average models. Moreover, a new residual index is proposed, which is insensitive to the mean changes in the process variables. The proposed method and dynamic improved partial least-squares (DIPLS [1]) technique are finally applied to a numerical case study and a non-isothermal continuous stirred tank reactor (CSTR) benchmark, and the simulation results demonstrate the effectiveness of the proposed scheme[2].

- **Chapter 4**

This chapter presents a new data-driven strategy for real-time root-cause fault diagnosis in (non-)linear processes by estimating the strength of causality using normalized transfer entropy (NTE) between measured process variables and variations of a residual signal. Moreover, for conducting causality analysis in the proposed strategy, a new and fast pathway for estimation of transfer entropy so-called symbolic dynamic-based normalized transfer entropy (SDNTE) is proposed. This chapter also introduces a new theory of joint xD-Markov machine to capture dynamic interactions between two time-series, based on which joint-Shannon entropies are utilized to develop the theory of SDNTE. The computational complexity of the proposed SDNTE is significantly lower than conventional kernel-based methods for estimating probability density functions (PDFs) in the calculation of TE. SDNTE also requires less amount of historical process data to reveal causality between two time-series, enabling early

[2] B. Rashidi, Q. Zhao, Quality-Related Fault Detection for Processes with Time-Varying Measurements, 2019 18th European Control Conference (ECC), 3873-3879

fault diagnosis and real-time application of transfer entropy. A part of this chapter is published in[3].

- **Chapter 5**

  This chapter proposes a general framework for autonomous root-cause fault diagnosis in a complex process. In this framework, as a prerequisite step after conducting fault detection, the potential root-cause fault candidates are selected using a contribution score-based method (e.g. accumulative rate contribution scores [2]). Then a fully automated procedure is proposed to determine the root-cause(s) of the detected fault amongst potential candidates without *a priori* knowledge of the base-line model or intervention of an expert. To locate the root-cause variable(s), firstly symbolic dynamic-based normalized transfer entropy (SDNTE) defined in Chapter 4 is used to generate an initial causal graph of root-cause fault candidates. Then symbolic dynamic filtering is further applied to estimate the DTE and symbolic dynamic-based normalized direct transfer entropy (SDNDTE) is proposed and utilized for pruning the initial graph (i.e. discard indirect and spurious causal edges). To this aim, explicit definitions of immediate intermediate variables (IIV) and source intermediate variables (SIV) are given and systematic algorithms are developed to find them efficiently. At last, a topological approach is proposed to autonomously locate the root-cause variables according to the pruned causal graph. To demonstrate the effectiveness and applicability, the proposed autonomous scheme is tested on a numerical example and finally validated on the Tennessee Eastman process (TEP) benchmark model.

[3] B. Rashidi, D. S. Singh, Q. Zhao, Data-driven root-cause fault diagnosis for multivariate non-linear processes, Control Engineering Practice, Elsevier, Volume 70, Pages 134-147, Nov 2017.

Chapter 2

Fault Detection of Non-Stationary Processes Using Moving-Mean PCA

2.1 Introduction

Fault detection in processes with non-stationary time-series measurements is a challenging line of research. In most of industrial processes, sensor measurements fall into the non-stationary category, in which the mean and/or variance vary in normal operating conditions. PCA has been recognized as a useful approach for fault detection in high-dimensional processes due to its simplicity and effectiveness. However, this approach suffers from a stationary assumption. To address this limitation, several modified versions of PCA (e.g. Adaptive PCA [36] [30] and moving window PCA [65]) are proposed such that upon receiving the test data subjected to non-stationary changes, the base-line model is updated using a set of algorithms. Although updating the base-line parameters using real-time adaptation can be a solution to address the non-stationarity, it introduces additional implementation limitations and adds computational complexity due to the real-time recalibration (e.g. conducting SVD). This motivated authors to propose a new PCA-based technique that not only handles the non-stationary changes in the time-series but also avoid a real-time updating structure to reduce the computational complexity and simplify the implementation. The core objective of this chapter is to distinguish between faults and process variations due to intentional/induced manipulated inputs changes. The proposed strategy can be applied to both stationary and non-stationary cases and provide feasible features about the health status of the process under study.

The following shows structure of the systems under study, in which the process measurements may have non-stationary statistic behaviour. Eq. (2.1) is a generic definition of a (non-)linear process such that the time-series $X \in \mathcal{R}^m$ are measured.

$$X = G(U) + w \tag{2.1}$$

where $U = [\nu, \mu]^T \in \mathcal{R}^{n+l}$. $\nu \in \mathcal{R}^n$ is the i.i.d noise and $\mu \in \mathcal{R}^l$ is the non-stationary time-series with time-variant mean, e.g. bounded random walk. w acts as the independent measurement noise and $G(.)$ can be a general linear or non-linear function. According to Eq.

(2.1), the mean and variance of the process variables might be subjected to changes under normal operating condition. However, this research only focuses on the mean-variation and assumes that the variance of the process variables remains relatively constant.

In this proposed approach, the normal operating condition is considered as non-stationary inevitable changes that domain experts expect as non-faulty process (non-)parametric changes such as set-point change, equipment degradation, etc. On the other hand, the fault scenarios that can be detected by the proposed approach includes but not limited to constant bias with different magnitudes and faults with deterministic trends such as ramp with slow and steep slope. In addition, stochastic random drift can also be handled by the proposed MM-PCA. To be able to distinguish between normal non-stationary mean changes and actual faults, it is assumed that the statistical dynamic of the fault scenarios are different from the normal non-stationary changes.

2.2 Moving-Mean PCA (MM-PCA) and Proposed Feature Indices

The proposed strategy includes two main steps; first, generating feature indices and, second, unsupervised probability distribution analysis to optimally distinguish the normal time-varying behavior of the process measurements (non-stationary) from actual faults. As a rule of thumb, for implementation, 30 percent of the entire available training data is used for learning the base-lines and calculating feature indices and 70 percent of that is utilized for learning a probability-based model for creating a fault hypothesis test. The following subsections present the details in each step of the proposed approach.

2.2.1 Feature Extraction

To conduct effective anomaly detection, three feasible feature indices are defined for a given process which carry key information about the non-stationary behaviour of process measurements. It should be noted that the formulation in this chapter is given for a linear process, but the non-linear extension of the proposed feature can be similarly derived by using KPCA (See *Appendix* for details of the KPCA method [34]).

The Zero-Order Error Index Φ^0_{MM}:

This index indicates the zero-order (constant) trend of a time-series according to the base-line model derived in the training step. To make this feature robust to time-varying mean changes of process variables, a moving-mean strategy is proposed as follows,

Step 1: A complete training data-set is collected to construct the process base-line model. If the process is assumed to be linear, dynamic PCA [23] with a proper number of augmentation shift h may be applied to reduce the dimensionality and extract the principal components used in the proposed framework. On the other hand, if the process is non-linear, kernel PCA [31] with a proper choice of kernel function (i.e. Gaussian, polynomial Sigmoid, etc.) can be utilized. As the first step, we determine the nominal average mean m_0 and average variance v_0 of the entire training data and accordingly conduct mean centering and standardize the time-series to unit variance.

Step 2: Apply PCA to calculate the proper transformation matrix $M_{\phi^0} = \dfrac{M_{T^2}}{UCL_{T^2}} + \dfrac{M_{SPE}}{UCL_{SPE}}$ (See *Appendix* for derivation of M_{SPE} and M_{T^2}), i.e. required for calculation of the combined index ϕ^0 [66], for the training data.

Remark 2.1 *The superscript* ^{0} *in the combined index* ϕ^0 *indicates the zero-order difference of the time-series, i.e. original signal with no differencing,* $X \in \mathcal{R}^{N\times m}$ *are used in PCA to derive the loading matrices.*

Step 3: Determine the combined index $\phi^0(i) = x(i)M_{\phi^0}x^T(i)$ using the baseline correlation matrix M_{ϕ^0} for a given new test measurement $x \in \mathcal{R}^m$.

The geometrical interpretation for the underlying concept of MM-PCA is to relocate the origin coordinates of the original multivariate signal space as long as their mean variations are within the normal/expected operating zone. This origin relocation maintains the stationarity assumption of PCA for transformation of the signal space to the scores while the signal mean varies. The upper bound of operating zone is defined according to the difference between combined index ϕ^0 and its threshold. To this end, it is proposed to define a moving window with a length of W_L in which weighted average filtering is conducted with respect to the difference of the combined residual index ϕ^0 from its nominal base-line threshold for the entire training data. As a result, the mean values of the variables used for mean centering get updated at each sample time and resist to exceed threshold as long as it is within the normal operating zone. However, if there exists a significant mode change due to malfunction occurring in the process which drives one/some of the variables out of the normal/expected operating zone, the mean values do not adapt to the fault induced changes, leading to successful detection of faults.

Step 4: Calculate the scalar distance $D(i) = \phi^0(i) - UCL_\phi$ between the current combined index ϕ^0 and its upper control limit $UCL_\phi = 2$ which is selected as a less conservative threshold based on the definition of combined index $\phi^0(i) = \dfrac{SPE(i)}{UCL_{SPE}} + \dfrac{T^2(i)}{UCL_{T^2}}$ (See *Appendix* for calculation of upper control limits).

Remark 2.2 *The upper control limit* UCL_ϕ *is considered as a tuning parameter in the proposed MM-PCA, which can be alternatively selected using the approximate distribution of* ϕ^0 *in [67]. When setting* $UCL_\phi = 2$*, it assumes that mean variations of the process measurements should be significant enough to distort both* T^2 *and* SPE *indices beyond their upper control limits for activating the proposed mean updating rule given in Eq. (2.2).*

Step 5: Use a switching function to activate the updating rule for the mean of the new test data,

$$
\begin{cases}
g_1(i) = 1, g_2(i) = 0 & D(i) < 0 \\
g_1(i) = 0, g_2(i) = 1 & D(i) \geq 0 \ and \ D(i) \leq \bar{\mathcal{V}} \\
g_1(i) = 1, g_2(i) = 0 & D(i) \geq \bar{\mathcal{V}}
\end{cases}
\tag{2.2}
$$

where $g_1(i)$ and $g_2(i)$ act as two switching parameters with respect to $D(i)$ to properly activate and deactivate the updating rule, as given in Eq. (2.3).

Step 6: Upon receiving a new test data $x(i)$, determine the updated mean $m^*(i)$ of the test data, which adjusts the original training data mean according to the normal/expected operating zone as follows,

$$
m^*(i) = g_1(i)m_0 + g_2(i) \sum_{j=0}^{W_L-1} \frac{W_L - j}{0.5W_L(W_L+1)} x(i-j) \tag{2.3}
$$

Remark 2.3 *The switching function in Eq. (2.2) is a simple choice for performing the mean recalibration. Other choices such as sigmoid or hyperbolic functions may be alternatively adopted to regulate the traversing action inside and outside of the normal operating zone (variation range $\bar{\mathcal{V}}$).*

Eq. (2.4) is another suggested form of the updating rule using a differentiable and smoother sigmoid function.

$$
\begin{aligned}
m^*(i) = & \left(\frac{2}{\left(\frac{1}{1+e^{-aD(i)}} - \frac{1}{1+e^{-a(D(i)-\bar{\mathcal{V}})}}\right) + 1} - 1 \right) m_0 \\
& + \left(\frac{1}{1+e^{-aD(i)}} - \frac{1}{1+e^{-a(D(i)-\bar{\mathcal{V}})}} \right) \sum_{j=0}^{W_L-1} \frac{W_L - j}{0.5W_L(W_L+1)} x^*(i-j)
\end{aligned}
\tag{2.4}
$$

where $a \geq 10$ is the tuning parameter to adjust the sharpness of the switching function. The above formulation normalizes the new test data using the updated mean and the same standard deviation. It should be noted that the standard deviation of the measurements is assumed almost constant.

In Eq. (2.2) and (2.4), the term $\bar{\mathcal{V}}$ stands for upper bound of variation of the normal operating zone which is also the upper bound of changes for combined index ϕ^0. This value can be calculated based on the operator's knowledge about the normal/expected range of variation of each process measurement. If the upper bounds of expected variations of all process measurements $\mathcal{V}_i \in \mathcal{V}_X, \ i = 1, ..., m$ are known, $\bar{\mathcal{V}}$ can be determined as $\bar{\mathcal{V}} = \mathcal{V}_X^T M_{\phi^0} \mathcal{V}_X$.

In practice, knowledge about the expected range of variations for all the process measurements might not be available. If the upper bound of expected variations of $a < m$ process measurements are known, it may still be possible to calculate the other $m - a$ unknown $\mathcal{V}_i$ using SVD as follows,

$$X_{N\times m} = \begin{bmatrix}\hat{U}_{N\times r} & \tilde{U}_{N\times(N-r)}\end{bmatrix} \begin{bmatrix}\hat{S}_{r\times r} & 0 \\ 0 & 0\end{bmatrix} \begin{bmatrix}\hat{V}_{m\times r} & \tilde{V}_{m\times(m-r)}\end{bmatrix}^T. \tag{2.5}$$

According to Eq. (2.5), $\tilde{V}_{m\times(m-r)}$ is the right *null* space of $X_{N\times m}$, which contains columns of V corresponding to the zero singular values. The auto-regressive relationship between process measurements $X = [x_1, x_2, ..., x_m]^T$ can be captured using the rows of $\tilde{V}_{m\times(m-r)}$ such that $\tilde{V}^T X = 0$. Using this homogeneous equation and knowing that $(m-a)$ upper bounds of measurements' variations are unknown, we can write $\tilde{V}^T \mathcal{V}_X = 0$, in which $\mathcal{V}_X$ is an array of all process measurements' upper bounds. The problem is then redefined and changed into solving a set of linear equations for $(m-a)$ unknown upper bounds. To this aim, by rearranging the columns of $\tilde{V}$ according to the rearranged upper bound matrix $\mathcal{V}_X = [\mathcal{V}_X^{unknown}\ ,\ \mathcal{V}_X^{known}]^T \in \mathcal{R}^m$, the matrix $A \in \mathcal{R}^{(m-r)\times(m-a)}$ is built such that the problem is redefined to solve the following,

$$A\mathcal{V}_X^{unknown} = C, \tag{2.6}$$

where $\mathcal{V}_X^{unknown} \in \mathcal{R}^{(m-a)}$. $C \in \mathcal{R}^{(m-r)}$ is calculated by multiplying $\mathcal{V}_X^{known}$ to the columns of the $\tilde{V}_{m\times(m-r)}$ corresponding to the a known upper bounds. The columns of $\tilde{V}_{m\times(m-r)}$ corresponding to the $(m-a)$ unknown upper bounds $\mathcal{V}_X^{unknown}$ are put together in matrix A. In general, Eq. (2.6) is consistent, i.e. it has at least one solution, if the row rank of augmented matrix $[A \mid C] \in \mathcal{R}^{(m-r)\times(m-a+1)}$ is equal to the row rank of coefficient matrix $A \in \mathcal{R}^{(m-r)\times(m-a)}$. This solution is unique if this rank is equal to $(m-a)$. Finally, when the upper bound of variations for all m process measurements are determined, the upper bound required in Eq. (2.2) is calculated as $\bar{\mathcal{V}} = \mathcal{V}_X^T M_{\phi^0} \mathcal{V}_X$.

Step 7: In the final step, as shown in Eq. (2.7), the zero-order moving-mean error index Φ^0_{MM} is defined as the first feature index that helps to capture the time-series behaviour.

$$\begin{aligned}\bar{X}_{test}(i) &= \frac{(X_{test}(i) - m^*(i))}{\nu_0} \in \mathcal{R}^m \\ \Phi^0_{MM}(i) &= \bar{X}_{test}(i) M_{\phi^0} \bar{X}_{test}(i)^T\end{aligned} \tag{2.7}$$

By following this updating rule for variables' means, it is assumed that the structure of the process is intact which implies that the principal directions remain the same during the process. Any relatively slow changes in the mean or oscillations due to normal operating variations are compensated by the adaptive mean rule. On the other hand, if there is a severe malfunction in the process which drives the combined index ϕ^0 to significantly exceed its threshold ($UCL_{\phi^0} = 2$), the combined index Φ^0_{MM} will not get updated and it will detect that malfunction.

The First-Order Error Index Φ_{MM}^1:

This index is defined to monitor the first order differencing (rate of change) of process variables. Although the non-stationary mean variations of the process variables are unexpectedly random, it is expected that the rate of change is bounded in many cases. For instance, in continuous stirred tank reactor (CSTR) process, there exists a catalyst that degrades along with time and induces a first-order (ramp) change into two variables during the operating process. In acetylene hydrogen reactor [48], some of the variables are subjected to drifting mean changes due to the degradation. Moreover, in the distillation column process, variables have a similar trend to a random walk signal, but the rate of change of the variations has a bounded envelope. The following shows the detailed steps to calculate the proposed Φ_{MM}^1:

Step 1: Use the nominal mean value m_0 determined in the first preprocessing step to bring all time-series to the comparable range to each other for PCA implementation.

Step 2: Consider a block-wise approach with the length of W_I, determine the average rate of change of the variables $X(i) \in \mathcal{R}^m$ as the following,

$$X^{d_1}(i) = \frac{1}{W_I}\left(\sum_{j=1}^{W_I} X(i-j) - \sum_{j=W_I}^{2W_I} X(i-j)\right) \tag{2.8}$$

Step 3: Conduct PCA on the $X^{d_1} \in \mathcal{R}^{N\times m}$ for N training samples to extract the principal transformation matrices M_{ϕ^1} by following the *Algorithm* 7 in the *Appendix.*

Step 4: determine the first-order error index as following,

$$\Phi_{MM}^1(i) = X^{d_1}(i) M_{\phi^1} {X^{d_1}}^T(i) \tag{2.9}$$

For a non-linear case, Kernel PCA will be applied on the time-series X^{d_1} to extract the kernel transformation matrices $M_{\phi^1}^{kernel}$ accordingly and the first order moving-mean feature can be similarly calculated as $\Phi_{MM}^1 = k(x^{d_1}) M_{\phi^1}^{KPCA} k(x^{d_1})^T$.

The Second-Order Error Index Φ_{MM}^2:

This index is defined to monitor the second-order differencing of the process variations. Similar to the aforementioned steps for determining Φ_{MM}^1, this index can also be determined accordingly. Using a block-wise approach, we obtain

$$X^{d_2}(i) = \frac{1}{W_I}\left(\sum_{j=1}^{W_I} X^{d_1}(i-j) - \sum_{j=W_I}^{2W_I} X^{d_1}(i-j)\right). \tag{2.10}$$

It should be noted that the training data X^{d_1} is already standardized before it is used to generate the training data for the second-order feature Φ_{MM}^2. After applying PCA on the

$X^{d_2} \in \mathcal{R}^{N \times m}$ for the linear case and deriving the corresponding transformation matrix, the proposed feature is determined as follows,

$$\Phi^2_{MM}(i) = X^{d_2}(i) M_{\phi^2} {X^{d_2}}^T(i) \tag{2.11}$$

Similarly for the non-linear case, it is defined as $\Phi^2_{MM} = k(x^{d_2}) M^{KPCA}_{\phi^2} k(x^{d_2})^T$, where $k(x^{d_2})$ represents the transformed test data with respect to *Algorithm* 8 in the *Appendix*.

Remark 2.4 *The first-order error index Φ^1_{MM} tends to monitor the rate of changes of the process time-series. Accordingly, the second-order error index Φ^2_{MM} monitors the acceleration of the changes. It should be noted that higher-order indices may be also derived and incorporated as additional features, but as a rule of thumb, the zero-, first- and second-order indices convey three feasible (mechanical) aspects of the variations in the processes under study.*

2.3 Unsupervised Non-Parametric Learning for Overall Index Derivation

In the previous section, three monitoring indices that carry useful information about the condition and variation behaviour of the process measurements are defined. In this section, it is proposed to utilize a probability-based approach to learn the normal behaviour of the features while the process is at normal (no-fault) operating condition. Fig. 2.1 illustrates the main steps in this section, in which an unsupervised probability-based learning method is utilized to distinguish non-stationary normal operating conditions form faulty counterparts without any need for pre-tagged training data from the domain experts. Assuming $X \in \mathcal{R}^{N \times m}$ with N observation is utilized for generating the feature indices, then, N_c is the number of sample point to build the feature matrix as follows,

$$X_f = [log(\Phi^0_{MM}), log(\Phi^1_{MM}), log(\Phi^2_{MM})] \in \mathcal{R}^{N_c \times 3} \tag{2.12}$$

Derivation of the feature indices $\Phi^n_{MM}, n = 0, 1, 2$ is presented in section 2.2.1. As shown in Fig. 2.2, the distribution shape of the feature indices are quite distinct from a normal *Gaussian* distribution. Therefore, logarithm trick is applied to transform the feature density function into a distribution shape which has more resemblance to the Gaussian distribution for further analysis.

Before using the matrix X_f, standardization is applied as a pre-processing step. After determining the proposed features, the goal is to generate an overall condition monitoring index and use it in the hypothesis testing for fault detection. In this case, a null hypothesis is defined for normal (no-fault) case to distinguish process malfunction from normal operating condition including non-stationary mean variations and normal mode changes. To achieve this, it is suggested to learn the probability density function(s) (PDF) of the feature indices for their normal operating condition. Therefore, for a given new process observation and its

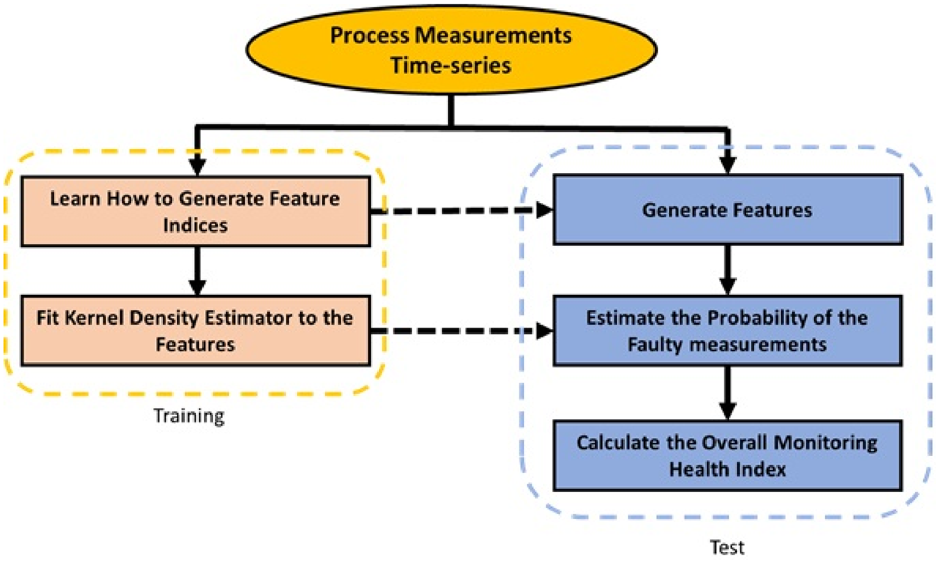

Process Measurements Time-series
Learn How to Generate Feature Indices
Fit Kernel Density Estimator to the Features
Training
Generate Features
Estimate the Probability of the Faulty measurements
Calculate the Overall Monitoring Health Index
Test

Figure 2.1: The flowchart of the training steps of the proposed framework.

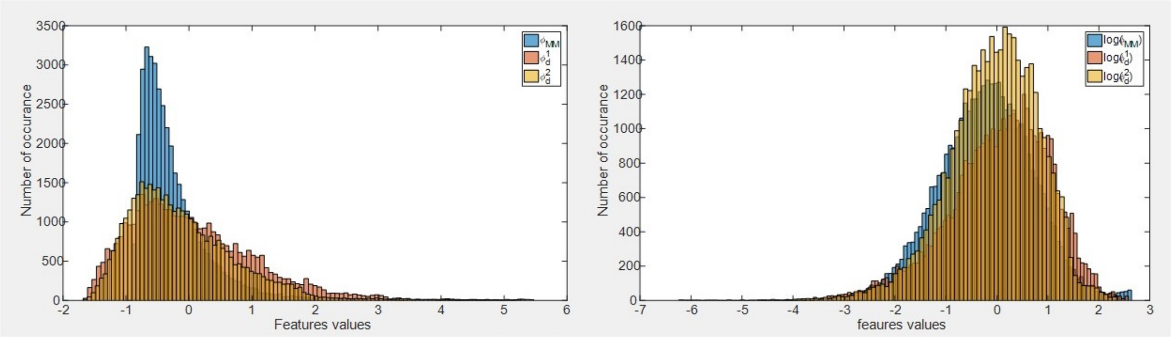


Figure 2.2: The histogram of the feature matrix defined in Eq. (2.12) with and without the logarithm trick for a synthetic numerical example in section 2.5.

corresponding feature indices, the estimated PDFs can be utilize to estimate how likely the new observation belong to the normal operating condition.

One conventional non-parametric approach for estimating the PDF of time-series is Kernel density estimator (KDE) [61]. We suggest to use KDE to approximate the individual PDF for each column of the feature matrix $x_f \in X_f$.

$$F_j^p(x_f^j) = \frac{1}{N_c h} \sum_{i=1}^{N_c} K(\frac{X - x_f(i)}{h}) \ \ , \ \ j = 1, 2, 3 \tag{2.13}$$

Where, $K(.)$ is a kernel function (e.g. Gaussian, spherical, Epanechnikov, etc.) satisfying Mercer's conditions. h is the bandwidth of the KDE which introduces a smoothing effect to the shape KDE. A large value of h leads to fitting a smoother kernel distribution function and a small value produces a sharper distribution curve with a higher level of fluctuations. Bandwidth h can be selected adaptively using the maximum likelihood method or k-nearest neighbor approach that updates h according to the Euclidean distance from the k^{th} nearest observation [68] [69].

For each column of the feature matrix X_f, an individual KDE is derived to estimate $\hat{P}_j(x_f(i))$ shown in Eq. (2.14) which stands for the estimated probability of feature index $x_j(i)$ such that it belongs to the corresponding normal operating condition.

$$\hat{P}_j(x_f(i)) = \int_{x_f(i)-\tau_j/2}^{x_f(i)+\tau_j/2} F_j^p(x)dx \simeq \tau_j F_j^p(x_f(i)) \ , \ j = 1, 2, 3 \tag{2.14}$$

To calculate the parameter τ_j in Eq. (2.14), first, the minimum and maximum values corresponding to the upper and lower 99.99% percentile of the $F_j^p(x_f^j)$ are estimated. Then according to the property of the density function such that $\int_{min(x_f^j)}^{max(x_f^j)} F_j^p(x)dx \simeq 1$, the interval between $min(x_f^j)$ and $max(x_f^j)$ can be divided to N_b subintervals. By using the Newton-Cotes formula, τ_j is approximated as $\tau_j = 1/\sum_{k=1}^{N_b-1} F_j^p(min(x_f^j) + k\frac{max(x_f^j) - min(x_f^j)}{N_b})$. When the process measurements are subjected to a fault, the feature indices will diverge from their normal operating conditions, hence, the estimated probability $\hat{P}_j(x_f^j(i)) \rightarrow 0$ depend on the severity of the fault-induced change. On the other hand under normal operating condition, each feature may be around the maximum possible probability of x_f^j which can be determined as $\gamma_j = \tau_j max(F_j^p(x_f^j(k))),\ k = 1, ..., N_b$.

After estimating $\hat{P}_j(x_f^j(i))$ for all three feature indices, they are utilized for calculation of the overall health index R as follows,

$$R(i) = a(1 - \frac{\Pi_{j=1}^3 \hat{P}_j(x_f^j(i))}{\Pi_{j=1}^3 \gamma_j}) \tag{2.15}$$

where a is a tuning parameter representing the upper bound of the overall index R.

Eq. (2.15) is designed to have certain desirable properties. For example, when there is a malfunction in the process and $\hat{P}_j(x_f(i))$ values are close to zero, the magnitude of overall health index reaches to the upper bound of $R = a$. This feature is favorable for industrial users because they mostly desire to work with a bounded health index which yields to the maximum for faulty condition and relatively negligible values for normal operating counterparts. Therefore, this health index is defined in such way to mostly generate values close to zero or its maximum limit.

As a final step, it is required to determine the upper control limit (UCL) for the proposed health index R for proper thresholding. For each feature index, the α tails percentile of the corresponding KDE is calculated and its corresponding value is considered as ϵ for that feature. In other words, for the j^{th} feature index, ϵ_j is determined as,

$$\epsilon_j = max(F_j^p(max(x_f^j)), F_j^p(min(x_f^j))) \;\; s.t. \;\; \int_{min(x_f^j)}^{max(x_f^j)} F_j^p(x)dx = \alpha \tag{2.16}$$

Hence, the overall UCL for the R in Eq. (2.15) is as follows,

$$UCL_R = a(1 - \frac{\Pi_{j=1}^3 \epsilon_j}{\Pi_{j=1}^3 \gamma_j}) \tag{2.17}$$

Upon observation of a new test process variables $x_f \in \mathcal{R}^3$, the proposed health index is determined by following the Eq. (2.15). Then the *null* hypothesis is defined as $H_0 : R < UCL_R$ (fault free), and the alternative hypothesis is $H_1 : R \geq UCL_R$ for faulty process. This means that if the overall health index R has values greater than its threshold, it supports the rejection of the *null* hypothesis.

2.4 Alarm-Based Process Monitoring

Alarm-based fault detection from the health residual signal in chemical processes is investigated in a great amount of research studies. In some cases, the end-user prefers binary alarm signal indicating whether the process is subjected to a malfunction. With this aim, an alarm generator can be utilized especially when the process is subjected to an oscillatory type of fault that leads to fluctuation in the proposed health index. Among results of alarm-based fault detection, some are focused on minimizing the fault detection delay [70] [71]. Although the proposed general health index in Eq. (2.15) has certain favorable advantages mentioned above, various uncertainties and disturbances such as occasional missing data, a surge in data acquisition system and sensor noise might create unwanted spikes that should not be detected as a process fault. To tackle this challenge, a rule-based alarm generator given in *Algorithm* 1 is proposed. The underlying idea in *Algorithm* 1 is based on 3-step processing of alarm signals using a moving average filter and alarm delay technique. First, a higher weight is assigned to the faulty residual samples to penalize the normal samples in comparison with faulty counterparts. Second, the length of the fault is considered to be greater than a predefined window to ensure that the alarm is not active for an outlier measurement or surge of DAQ card due to digitization. Finally, a rule-based approach is deemed to connect the

Algorithm 1 Alarm generating procedure using the proposed monitoring index R

```
INPUTS of Algorithm:
        $R(i)$ := The overall health index
        $UCL_R$ := The upper control limit of overall health index
        $b$ := Weighting parameter for marking up the faulty observations
        $w_1$ := Window size for weighted averaging of overall health index
        $w_2$ := The window size for fault continuity test
(Assign more weights to the residuals samples greater than UCL)
if $R_c(i) < UCL_R$ then
    $R_c(i) = R(i)$
else
    $R_c(i) = bR(i)$
(Define a window of length $w_1$ to store previous weighted $R_c$)
$R_1(i) = \frac{1}{w_1} \sum_{j=0}^{j=w_1} R_c(i-j)$
——————— (First Layer Alarm Generator → $Alarm_1$) ———————-
if $R_1(i)) > UCL_R$ then
    $Alarm_1(i) = True$
else
    $Alarm_1(i) = False$
——————— (Second Layer Alarm Generator → $Alarm_2$) ———————
if $Alarm_1(i) == True$ then
    if $\frac{1}{w_2} \sum_{j=i-w_2}^{j=i} Bool_{val}(Alarm_1(j)) > 0.75$ then
        $Alarm_2(i) = True$
    else
        $Alarm_2(i) = False$
else
    $Alarm_2(i) = False$
——————— (Third Layer Alarm Generator → $Alarm_3$) ———————
if $Alarm_1(i) == True$ & $Alarm_2(i-1) == True$ then
    $Alarm_3(i) = True$
else if $Alarm_1(i) == True$ & $Alarm_2(i) == False$ & $Alarm_3(i-1) == True$ then
    $Alarm_3 = True$
else
    $Alarm_3(i) = False$
OUTPUT of Algorithm: $Alarm_2 \Rightarrow$ Caution, $Alarm_3 \Rightarrow$ Fault

```

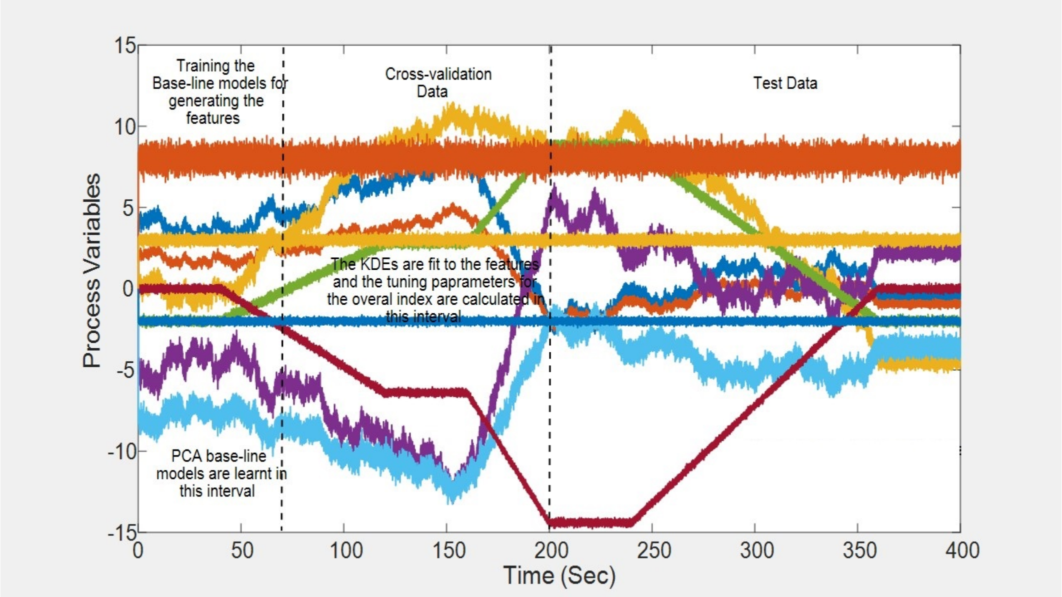


Figure 2.3: The time-series of the synthetic numerical example for different normal operating modes.

entire faulty zone and create a continuous alarm for the detected malfunction. In addition to the final fault alarm ($Alarm_3$), *Algorithm* 1 also generates a caution signal ($Alarm_2$) to notify operators to consider proper maintenance actions before the process reaches the more severe faulty condition.

2.5 Simulation Results

To verify the performance of the proposed fault detection scheme, at first, a synthetic numerical example is given and simulation results are shown in this section. The following shows the process model initially identified in Eq. (2.1),

$$X = UP + w \quad , \quad P = \begin{bmatrix} 3 & 2 & 3 & -5 & 0 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & -1.5 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & -3 & 0 & -2 & -5 & 0 & -2 & 8 & 3 \end{bmatrix} \tag{2.18}$$

where $U = [\nu, \mu]^T \in \mathcal{R}^3$, $\nu \sim \mathcal{N}(0, 0.05), \delta \sim \mathcal{N}(0, 0.005)$ and $w \sim \mathcal{N}(0, 0.0005) \in \mathcal{R}^{10}$. Also, $\mu(i) = \mu(i-1) + \delta(i-1)$ acts as a random walk noise that introduce the non-stationary behaviour to the process measurements. In Fig. 2.3, the process variables of the above synthetic simulation are illustrated. The process is simulated for 400 seconds with sample time T=0.01 second, which yields N=40,000 observations. The entire simulation composes 4 different modes in which only two of them are considered in the training base-line features and KDE determination and the other two modes are unknown to the proposed framework during the learning process. The expected upper bound of the normal variation range of the time-series are known as $\mathcal{V} = [10, 7, 14, 19, 22, 10, 14, 15, 2, 3, 3]^T$. then $\bar{\mathcal{V}} = \mathcal{V} M_{\phi^0} \mathcal{V}^T = 298$.

When training the base-line models while applying PCA, the cumulative percentage variance $CVP = 95\%$ is chosen for all features. Fig. 2.4(a) shows the indices Φ^0_{MM} and ϕ^0 (i.e. combined index from DPCA [66]) for the normal operating condition and shows how the proposed moving-mean index is robust to the mode change and non-stationary variations. After determining the feature matrix X_f in Eq. (2.12) for this process, the histogram of each column of the matrix after and before conducting the logarithm trick is shown in Fig. 2.2. The objective of this simulation study is to demonstrate whether the proposed framework and overall health index R can be used to effectively distinguish the normal operating mode changes or non-stationary variations from the actual process malfunctions. In Fig. 2.4(b), the overall index is shown for normal operating condition and it can be observed that while there are two significant mode changes and non-stationary mean variations of four process variables, the overall index can successfully flag normal operating condition. In the next step, three different fault scenarios are defined and introduced to the process to evaluate the capability of the proposed framework in detecting faults. It is worth mentioning that all the fault scenarios chosen in the simulation should satisfy the detectability criteria [68].

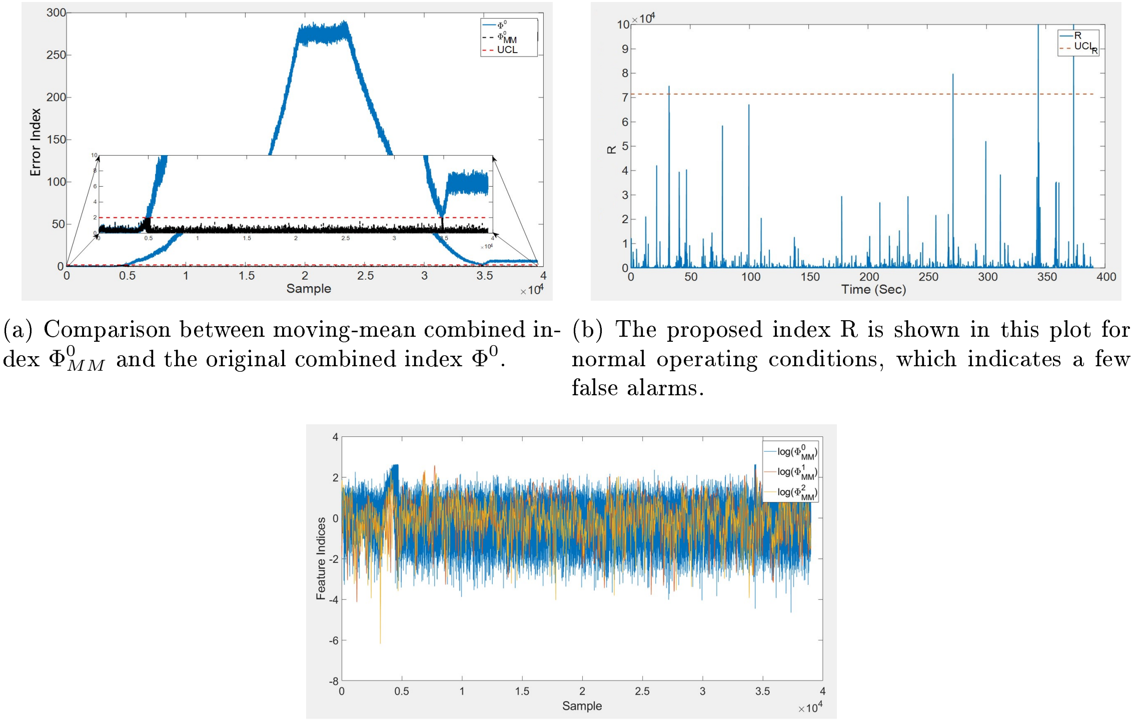
The proposed MM-PCA approach is mainly designed to perform fault detection in both stationary and non-stationary processes. This means that the process operators do not need to manually monitor the stationarity of the process time-series and the proposed approach performs as an ordinary (kernel) PCA when non-stationary changes do not exist.

2.5.1 First Fault Scenario: Bias

An additive bias fault $F = [0, 8, 0, 0, 0, 0, 0, 0, 0, 0]^T$ is added at 290^{th} second to the process and overall health index R is shown in Fig. 2.5(b). This indicates that although the magnitude of the additive fault is in the safe range of variation for the second variable, but the SPE portion of the combine index ϕ^0 reacts aggressively and exceeds $\bar{\mathcal{V}} = 298$. Therefore, as shown in the Fig. 2.5(a), the mean updating rule is deactivated and the error index Φ^0_{MM} shows the fault .

2.5.2 Second Fault Scenario: Slow Ramp Variation

In this case, an additive slow ramp variation with the slope of 0.05 is introduced to the x_{10} at 290^{th} second. The challenge in this scenario is that the additive fault does not drag the process measurements out of their upper bounds, hence as shown in Fig. 2.6(a), the moving-mean updating rule is active and Φ^0_{MM} does not reject the *null* hypothesis and it is blind to the fault. For this type of fault scenarios, the first-order error index Φ^1_{MM} plays a prominent role in fault detection. This scenario appears as a bias for the first-order difference of the time-series. Fig. 2.6(b) shows R for this scenario which successfully detect the presence of the malfunction.

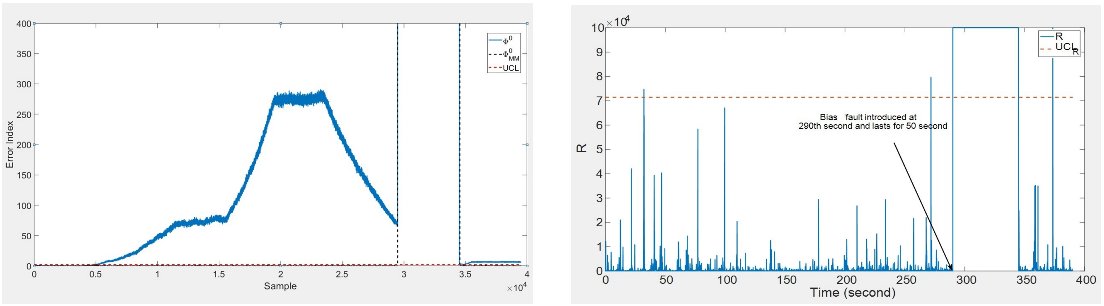


(a) Comparison between moving-mean combined index Φ^0_{MM} and the original combined index Φ^0.

(b) The proposed index R is shown in this plot for normal operating conditions, which indicates a few false alarms.

(c) The feature indices X_f after normalization.

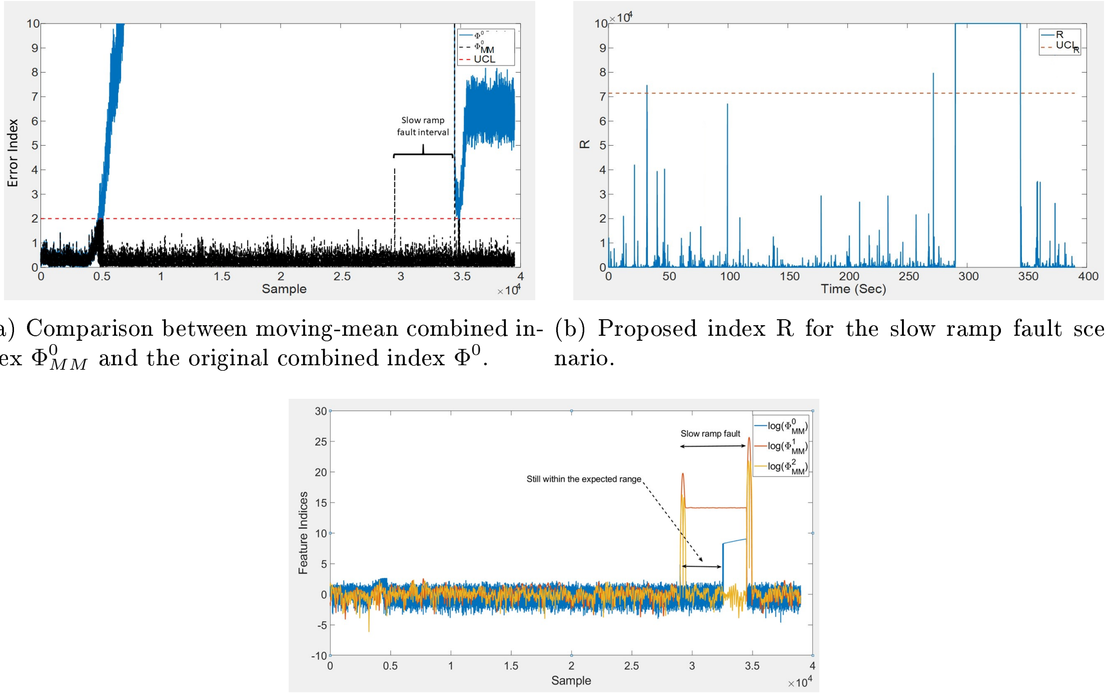
Figure 2.4: The error indices using MM-PCA and the DPCA approach in the numerical example under normal operating condition.



(a) Comparison between moving-mean combined index Φ^0_{MM} and the original combined index Φ^0.

(b) Proposed index R.

Figure 2.5: The general health index R for the first fault scenario (additive bias).



(a) Comparison between moving-mean combined index Φ^0_{MM} and the original combined index Φ^0.

(b) Proposed index R for the slow ramp fault scenario.

(c) Feature indices X_f after normalization.

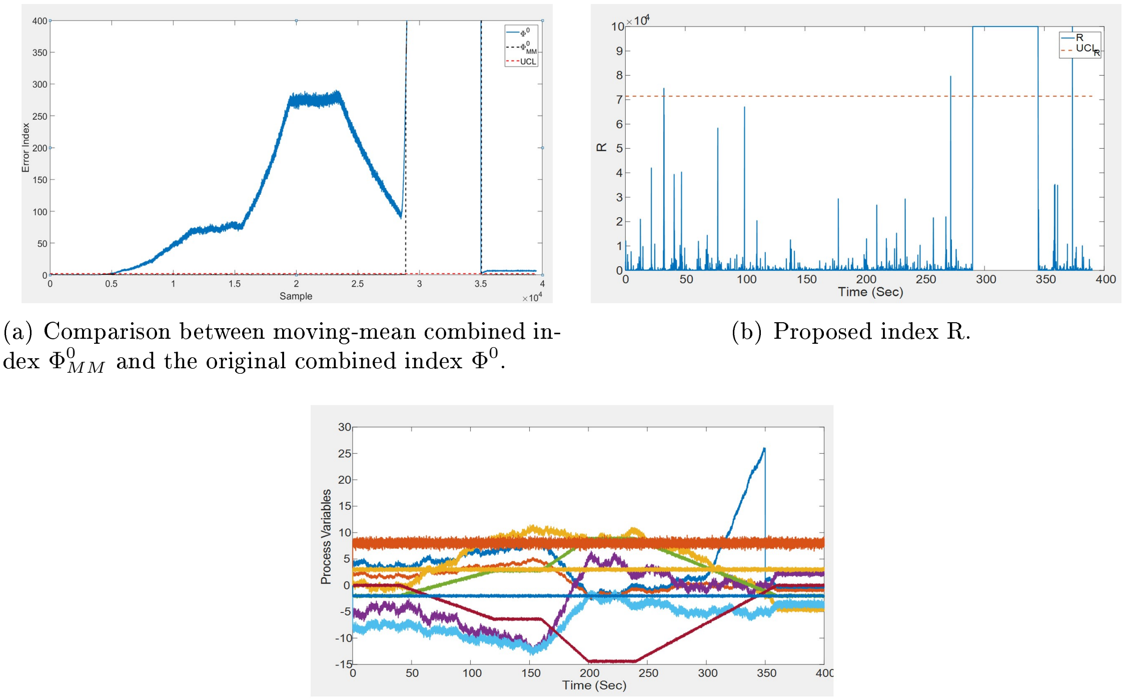
Figure 2.6: The comparison between the proposed error indices using moving-mean PCA and the DPCA approach in the second fault scenario (slow ramp).

2.5.3 Third Fault Scenario: Steep Ramp

In this case, x_3 and x_6 are subjected to a fault with steep ramp trend with -0.2 and 0.4 slopes, respectively, which drag mean variations out of the expected range of time-series $\bar{\mathcal{V}}$ as shown in Fig. 2.7(c). Both Φ^0_{MM} and Φ^1_{MM} can detect this fault and R (i.e. presented in Fig. 2.7(b)) could successfully flag the fault which can be considered as an obvious fault scenario for the proposed framework.

2.5.4 Fourth Fault Scenario: Random Drift

In this case, a random drift $d(i) = d(i-1) + \epsilon(i-1)$, $\epsilon \sim \mathcal{N}(0, 0.1)$, is added to the x_4 at 300^{th} second and the time-series are shown in Fig. 2.8(c). As can be seen in Fig. 2.8(a), the index ϕ^0 fluctuates around the upper bound $\bar{\mathcal{V}} = 298$ which results in false alarms in moving-mean zero-order index Φ^0_{MM}. The challenge of the random drift fault is that it does not have a deterministic trend that can be detrended by regression-based approaches. Also, random drift malfunction frequently drags the process measurements' mean to cross their upper bounds $\bar{\mathcal{V}}$ randomly and increase the false alarm rate if using only the zero-order index Φ^0_{MM}. On the other hand, as shown in Fig. 2.8(b), the combined index R could successfully detect the fault with no false alarm due to feature indices Φ^1_{MM} and Φ^2_{MM}, for which their normalized log values are shown in Fig. 2.8(d).



(a) Comparison between moving-mean combined index Φ^0_{MM} and the original combined index Φ^0.

(b) Proposed index R.

(c) Process measurements

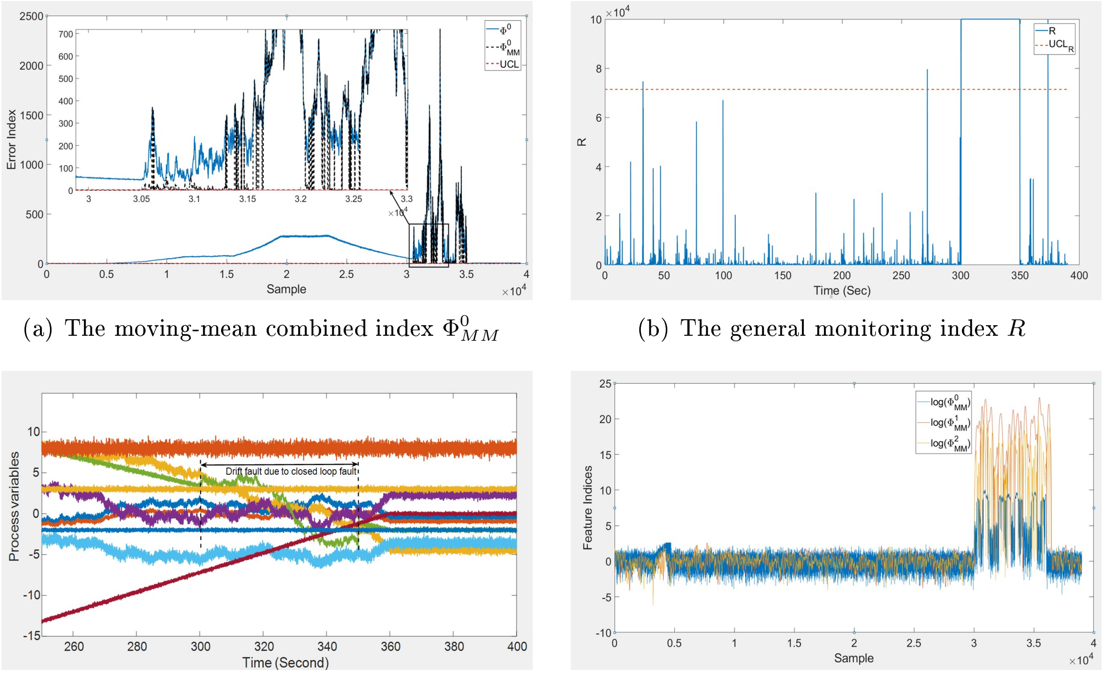
Figure 2.7: Simulation result of numerical example for third fault scenario (steep ramp).

Table 2.1: Sensor measurement classification for industrial compressor; each class belongs to a different component.

Vibration Measurement	Sealing System	Lubrication Oil	Process Measurement
$x_1 \rightarrow x_9$	x_{10}, x_{11}, x_{12}	x_{13}, x_{14}, x_{15}	$x_{16} \rightarrow x_{26}$

2.6 Industrial Application

An industrial compressor data set including 26 sensor measurements categorized in Table 2.1 is investigated for validating the performance of the proposed strategy. The main reason for considering this industrial application is that the compressor process is generally a complex non-linear and time-varying system, and the measurement data has regular non-stationary mean variations, which is the case of interests for this research. In this industrial application, only the process variables x_{16} to x_{26} are considered for monitoring the process health condition. From inspection and prior knowledge of domain experts, some normal batches of data are collected and used for training the non-linear base-line model and extracting the proposed features. Sampling rate for all measurement is $1 sample/min$. The indices of the training data batches are 1000 to 2000, 4500 to 5500, 15000 to 15500, 26000 to 27000 and 96000 to 97000. The length of weighted moving average filter in Eq. (2.3) $W_l = 10$ and for generating the first- and second-order feature indices shown in Eqs. (2.9) and (2.11) , a window of size $W_I = 5$ is considered acting as a moving average filter on the measurement increments. The



(a) The moving-mean combined index Φ^0_{MM}

(b) The general monitoring index R

(c) The process variables for the fourth fault scenario (random drift)

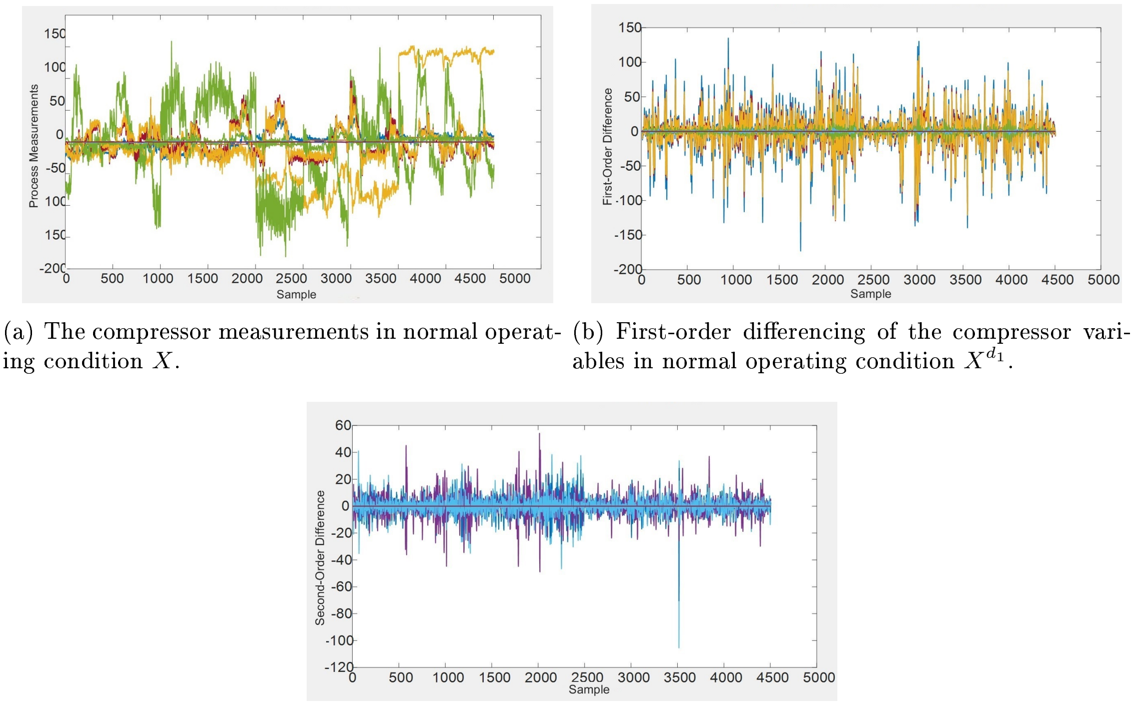
(d) The feature indices X_f after normalization for the fourth fault scenario (random drift)

Figure 2.8: Simulation result of numerical example for fourth fault scenario (random drift).

relationships among the compressor measurements follow a non-linear structure, hence, KPCA is adopted, in which RBF kernel function is used. To decompose the principal direction of the kernel matrix for the entire three features, $CVP = 98\%$ is considered and the number of principal components is determined to be 5, 1 and 4 for the three feature indices, respectively. Fig. 2.9 shows the standardized training data used for KPCA in training mode. Also, the first and second-order rate of variations for extracting the feature indices are shown in Figs. 2.9(b) and 2.9(c), respectively.

Fig. 2.10 presents the result KPCA fault detection in the compressor data utilizing the proposed moving-mean method alone with the *Algorithm* 1 for generating caution and alarm signals. The red spot (i.e. circles) represents the fault alarm, and the purple star shows the caution signal and the status of the process should be investigated. The idea behind generating a caution flag is to account for a situation that a fault at its primitive stage is developing or a disturbance case due to data acquisition surge, sensor spikes (i.e. oscillation) and missing data.

Moreover, Fig. 2.10 demonstrates advantages of the proposed moving-mean concept Φ^0_{MM} in Eq. (2.4) for updating the mean of the variables in expected normal operating range. On the other hand, Fig. 2.11(c) shows the proposed health index R for the compressor data by fitting a multivariate kernel density estimator. As can be seen, the health index is mostly zero for the normal condition and is maximum for the faulty periods. This attribute of the health index helps the operators to compartmentalize the normal condition from anomalies much



(a) The compressor measurements in normal operating condition X.

(b) First-order differencing of the compressor variables in normal operating condition X^{d_1}.

(c) Second-order differencing of the compressor variables in normal operating condition X^{d_2}.

Figure 2.9: The zero-order, first-order and second-order difference of the time-series for the industrial compressor training data-set.

easier comparing to similar methods. Although the manipulated inputs of the compressor in the range of 30,000 to 40,000 induce a mean change into entire process variables, the health index could successfully recognize that as a normal operating variation and stays within the threshold. It should be noted that there exist several spikes and outlier samples of the proposed health index around the upper control limit which are mainly due to the presence of uncertainties, disturbances or missing data. This issue can be handled by applying an alarm generator such as the one proposed in *Algorithm* 1. The zoomed snapshots of two evens are also illustrated in Figs. 2.11(a) and 2.11(b) which indicate the response of overall index R with higher resolution.

2.7 Summary

The first yet important step for a thorough process health monitoring is to detect the presence of malfunction(s). Although there are different data-driven methods for fault detection, they mostly suffer from stationarity assumptions. One of the frequently used methods is principal component analysis (PCA), which assumes that the process time-series follow *Gaussian* distributions with time-invariant mean and variance. However, this assumption is violated in the majority of the industrial processes such as chemical plants and reactors, e.g. continuous

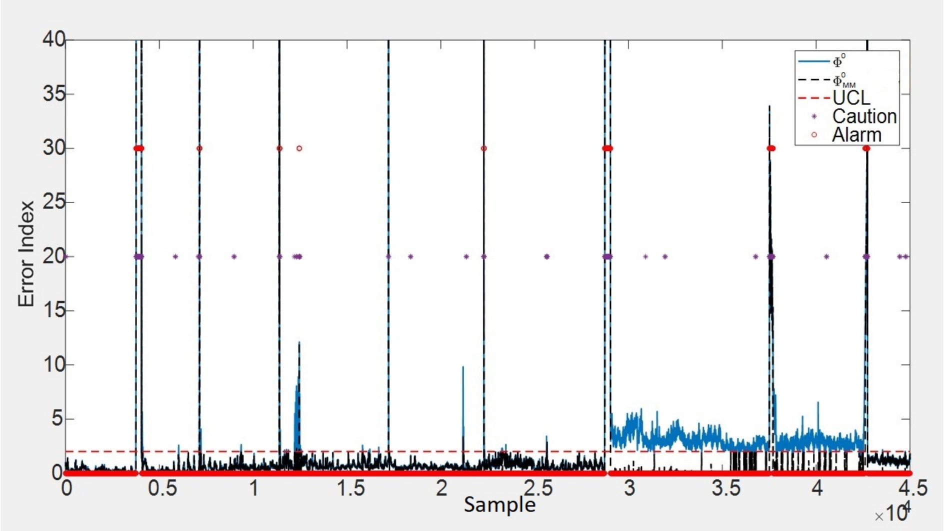
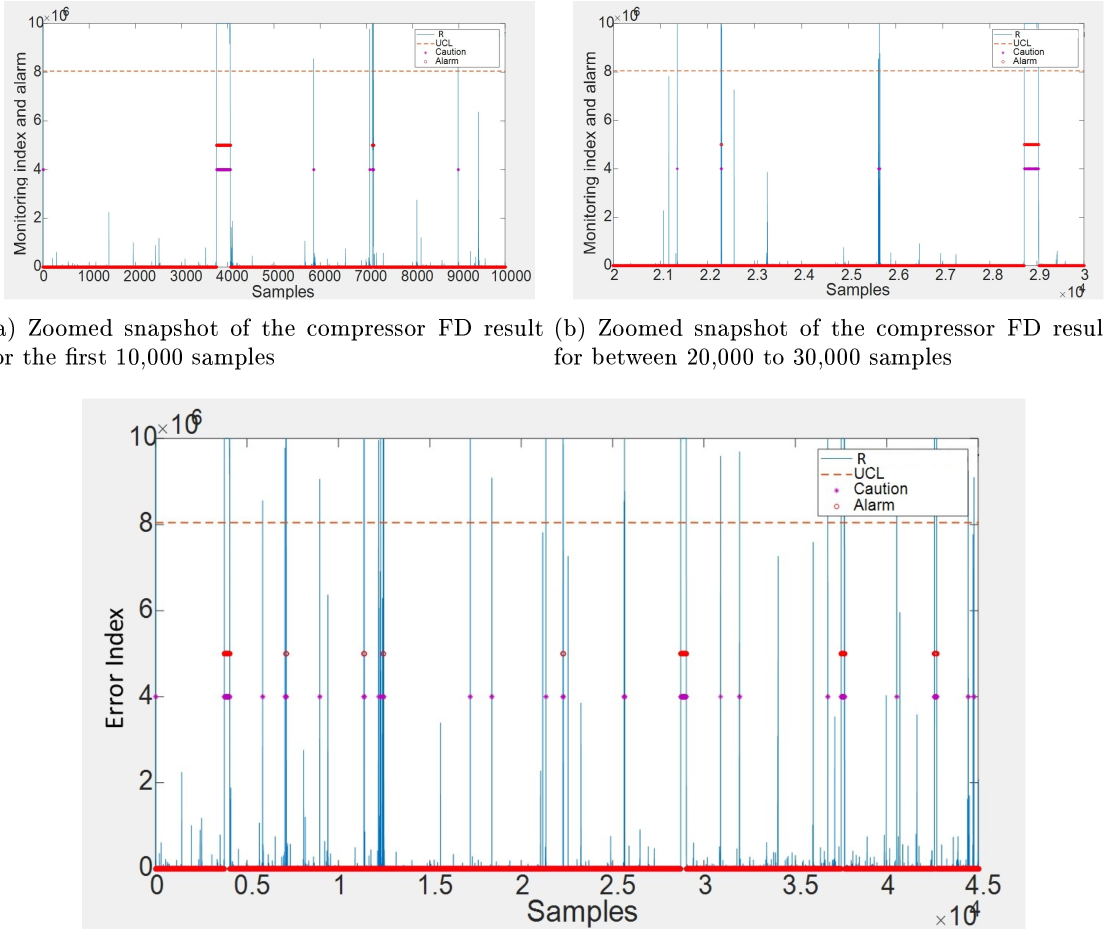


Figure 2.10: The generated alarm log-history using Algorithm 1 for 5 months of the compressor data. This figure compares the difference of the proposed MM-KPCA and conventional kernel PCA.

stirred tank reactor (CSTR), compressor in power plant, etc. The non-stationary nature of the time-series may be due to the presence of manipulated inputs, system degradation, and close-loop compensation action of the controllers.

In this chapter, a moving-mean PCA (MM-PCA) method is proposed that is applicable for both stationary and non-stationary processes. The basis of the proposed MM-PCA is to update the mean value of the variables based on their upper bound of expected range of variations. New feasible feature indices are defined which are good indicators for the statistical behavior of the process variable. Moreover, a non-parametric approach based on a kernel density estimator is used to generate a new health index to reconcile the features. In the end, an alarm-based algorithm is proposed to generate caution alarm and fault alarm to make the proposed method applicable to the industrial chemical process. The effectiveness of the proposed approach is presented in a numerical synthetic example for different fault scenarios. A real industrial compressor is also used to show the industrial implication of the proposed strategy for a non-stationary process subjected to time-variant mean changes of the measurement time-series.



(a) Zoomed snapshot of the compressor FD result for the first 10,000 samples

(b) Zoomed snapshot of the compressor FD result for between 20,000 to 30,000 samples

(c) The proposed health index R for the industrial compress

Figure 2.11: Process monitoring result of the compressor data using proposed framework.

Chapter 3

PLS-Based Quality Output-Related Process Monitoring in Non-Stationary Processes

3.1 Introduction

Amongst model-based and data-driven approaches, the latter has attracted considerable attention for quality-related process monitoring due to their distinct advantages of easy implementation in high-dimensional processes and less requirement for process knowledge. Although quality output-related fault detection has been extensively studied using PLS-based approaches [26] [42] [1], they usually assume that the process measurements have a stationary statistic behavior. To handle the cases with time-series subjected to time-variant changes, adaptive/recursive PLS solutions [47] have been applied to update the base-line model upon receiving a new batch of data. The updating mechanism for this solution induces a real-time analysis with a relatively high computational complexity that might hinder the industrial applications.

Another solution to monitor the non-stationary processes subjected to variations which change the process base-line structure is to create a bank of models for each mode (scenario) in the training phase and according to proper classification, apply the corresponding model for a certain batch of given test data. This solution is recognized as a supervised fault detection technique which is different from the proposed approach in this chapter. In this chapter, a cascade approach is proposed to overcome the time-variation of operating point(s) when process variables have a dynamic relation with quality-outputs (i.e. key performance indicators (KPIs)). According to the regression relationship found by dynamic improved PLS technique [1], first, the process variables are orthogonally decomposed into quality output-related and quality output-unrelated subspaces, second, it is proposed to obtain a principal manifold for each individual subspace, which remains unchanged during the normal time-varying operating mode. In addition, new residual statistics and logics are developed to successfully monitor quality output measurements. The proposed PLS-based approach is considered as an alternative for adaptive and recursive PLS counterparts [47] with simply less calculation complexity and providing the same fault detection result. The improvement in the proposed approach is that the online mean adjusting step is replaced with the off-line training of principal manifolds to capture the underlying structure of the output-relevant and irrelevant measurements.

3.2 Preliminaries and Problem Statement

For an industrial process with n measurements and m process quality output measurements (KPI), data matrices $X \in \mathcal{R}^{N\times n}$ and $Y \in \mathcal{R}^{N\times m}$ with $N \gg n > m \geq 1$ observations are built for process health monitoring. For this case, we assume that the underlying relationship between X and Y is linear, and the process measurements time-series have a normal distribution with a time-variant mean and constant standard deviation as the following,

$$x \sim \mathcal{N}(m_x(t), \Sigma_x), \quad y \sim \mathcal{N}(m_y(t), \Sigma_y) \tag{3.1}$$

The mean variations of the time-series might be the result of closed-loop control actions for dissipating disturbances, equipment degradation/corrosion, the evolution of malfunction(s) with time in the structure of the process equipment, intentional variation of the set-points and manipulated inputs, etc.

Before any further analysis, columns of X and Y should be normalized using their nominal mean values to bring their variations to a comparable range. In addition to the aforementioned assumptions, it is also assumed that the quality outputs Y are dependent on both current and past values of the X, hence, has a dynamically linear relation with them. In order to incorporate this dynamic interconnection, we assume that the dynamic dependency order of h between X and Y is known, thus, the augmented version of the process measurements $X_A \in \mathcal{R}^{(N-h)\times(nh+n)}$ is constructed as follows,

$$X_A = \begin{bmatrix} x_{h+1}^T & x_h^T & ... & x_1^T \\ x_{h+2}^T & x_{h+1}^T & ... & x_2^T \\ ... & ... & ... & ... \\ x_N^T & x_{N-1}^T & ... & x_{N-h}^T \end{bmatrix}, \qquad Y = \begin{bmatrix} y_{h+1}^T \\ y_{h+2}^T \\ ... \\ y_N^T \end{bmatrix} \tag{3.2}$$

In Eq. (3.2), $Y \in \mathcal{R}^{(N-h)\times m}$ is also re-defined such that its dimension matches with X_A. The above augmentation step converts the underlying dynamic relation between X and Y to a static linear counterpart $Y = X_A\psi + W$. Therefore, the problem is simplified to finding a regression model for a linear static process.

Improved PLS (IPLS) was recently proposed for quality output-related fault detection [1] [43]. To particularly monitor the impact of malfunctions on the quality outputs, a complete orthogonal decomposition is done on process measurement X_A at first,

$$\begin{cases} X_A & = \hat{X}_A + \tilde{X}_A \\ Y & = \hat{Y} + \tilde{Y} \Rightarrow Y = X_A\psi + \tilde{Y} \end{cases} \tag{3.3}$$

where,

$$\psi = ({X_A}^T X_A)^{\dagger} {X_A}^T Y \ \in \ \mathcal{R}^{n(h+1)\times m}. \tag{3.4}$$

If ${X_A}^T X_A$ in Eq. (3.4) is not full-rank, SVD can be used to calculate the pseudo inverse. Assume that X_A is decomposed in a way that $\hat{X}_A$ includes all the information in Y, hence $\tilde{X}_A$ should be uncorrelated with Y and perpendicular to $\hat{X}_A$. For this purpose, correlation matrix ψ in Eq. (3.4) is utilized to perform the orthogonal decomposition of X_A. By conducting SVD on $\psi\psi^T$, loading matrices required for projecting X_A onto $\hat{X}_A \in span\{\psi\}$ and $\tilde{X}_A \in span\{\psi^{\perp}\}$ are derived by first performing SVD on $\psi\psi^T$ as following,

$$\psi\psi^T = \begin{bmatrix} \hat{\Gamma}_\psi & \tilde{\Gamma}_\psi \end{bmatrix} \begin{bmatrix} \Lambda_\psi & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\Gamma}_\psi^T \\ \tilde{\Gamma}_\psi^T \end{bmatrix} \tag{3.5}$$

where, $\hat{\Gamma}\psi \in \mathcal{R}^{n(h+1)\times m}$, $\tilde{\Gamma}_\psi \in \mathcal{R}^{n(h+1)\times(n(h+1)-m)}$ and $\Lambda_\psi \in \mathcal{R}^{m\times m}$. Thus, the projection matrices are obtained as $\Pi_\psi = \hat{\Gamma}\psi\hat{\Gamma}\psi^T$ and $\Pi_\psi^{\perp} = \tilde{\Gamma}_\psi\tilde{\Gamma}_\psi^T$, as a result, the $\hat{X}_A$ and $\tilde{X}_A$ are determined as,

$$\begin{aligned} \hat{X}_A &= X_A \Pi_\psi \in \mathcal{R}^{(N-h)\times n(h+1)} \\ \tilde{X}_A &= X_A \Pi_\psi^{\perp} \in \mathcal{R}^{(N-h)\times n(h+1)} \end{aligned} \tag{3.6}$$

After successfully projecting the process variables onto two quality output-related and quality output-unrelated components using Eq. (3.6), for a given new process variables $x_{new} \in \mathcal{R}^n$, the quality output-related and output output-unrelated scores $\hat{t}_{x_{new_A}} = x_{new_A}\hat{\Gamma}_\psi$ and $\tilde{t}_{x_{new_A}} = x_{new_A}\tilde{\Gamma}_\psi$ are obtained, respectively. Since the score matrices $\hat{T}_{X_A} = \hat{X}_A\hat{\Gamma}_\psi \in \mathcal{R}^{(N-h)\times q}$ and $\tilde{T}_{X_A} = \tilde{X}_A\tilde{\Gamma}_\psi \in \mathcal{R}^{(N-h)\times n(h+1)-q}$ have full column rank q and $m(h+1)-q$, respectively, the Hotelling T^2 statistics can be utilized to monitor them as follows [43] [1],

$$\begin{aligned} \hat{T}^2_{X_A} &= \hat{t}^T_{x_{new_A}} \left(\frac{\hat{T}^T_{X_A}\hat{T}_{X_A}}{(N-h)-1} \right)^{-1} \hat{t}_{x_{new_A}} \\ \tilde{T}^2_{X_A} &= \tilde{t}^T_{x_{new_A}} \left(\frac{\tilde{T}^T_{X_A}\tilde{T}_{X_A}}{(N-h)-1} \right)^{-1} \tilde{t}_{x_{new_A}} \end{aligned} \tag{3.7}$$

Given a process, when the mean of one or some of the measurements X_A is subjected to variation under the normal operating condition, both of the quality output-related and quality output-unrelated T^2 statistics in Eq. (3.7) may violate their upper control limits. The reason behind this issue is that this conventional T^2 index particularly measures the distance of the scores' mean-values from the origin of X_A coordinate. For instance, as demonstrated in Fig. 3.1, for a hypothetical process with two scores $\hat{t}_{X_A} \in \mathcal{R}^{N\times 2}$, although the process is still in its normal operating condition, T^2 index measures each data-point's distance from the origin and can falsely report presence of a fault if the mean-values of process measurements change. In general, application of Eq. (3.7) is valid only when mean-values of the X_A's columns, and the

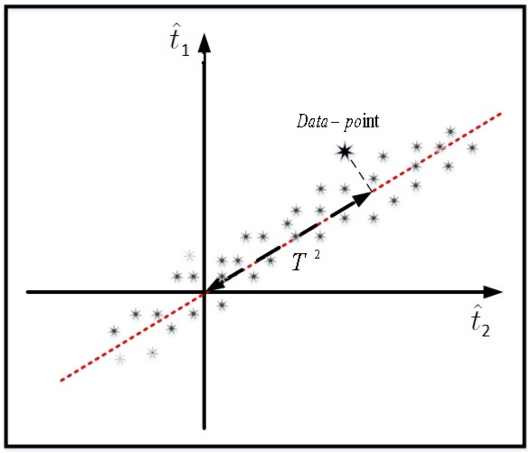


Figure 3.1: The interpretation of the T^2 index defined on scores in Eq. (3.7). T_1 and T_2 are two scores derived for this hypothetical example.

columns of score matrices $\hat{T}_A$ and $\tilde{T}_A$ are consistently around zero after mean-centering. This condition originates from stationarity assumption of the PLS approach, which fails in quality output monitoring of those processes in which their measurement time-series are subjected to non-stationary mean changes.

Rather than supervised classification-based solutions to resolve this shortcoming [45] [46], there exist two other options. One is to proceed with relatively computationally intense adaptive/recursive methods utilized in [47] to recalibrate the base-line models and thresholds in the online phase. The other approach is to propose a set of post-processing procedures without any base-line recalibration in the online phase and capture the structure of normal changes in the training phase. The latter approach is adopted in this chapter and a cascade PLS-based monitoring scheme is proposed which monitors the underlying principal manifold. The suggested manifolds are constructed to preserve the dynamic linear relationship between the principal loading vectors of $\hat{X}_A$ and $\tilde{X}_A$. The proposed scheme can be considered as the modified version of the method proposed in [1] [43], which can be applied for general dynamic linear processes for which the measurement may vary with time or remain consistent during the normal operating conditions. Not only this approach can be used for non-stationary processes, but it also has low computation cost because it requires no online updating/calculation for the adaptation of base-line parameters.

3.3 Proposed Method for Quality Output-Related Fault Detection

After building the regression structure shown in Eq. (3.3) and decomposing X_A to $\hat{X}_A$ and $\tilde{X_A}$, it is proposed to define a principal manifold for each component. This new modification makes the PLS algorithm robust to the normal changes in the operating condition since it does not measure the absolute distance of the scores from the origin but instead measures the

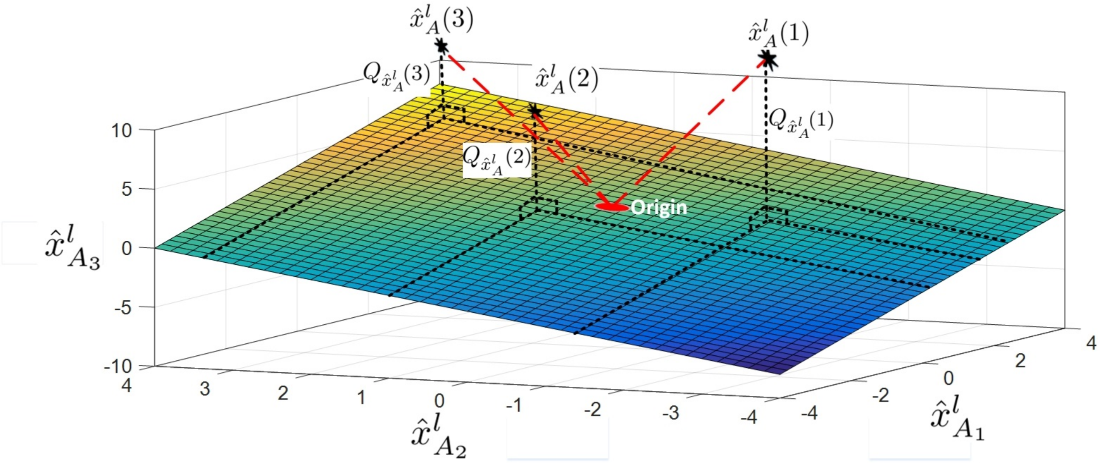


Figure 3.2: A pictorial example with three dimensional $\hat{X}^l_A$ and geometrical overview of proposed residual in Eq (3.9). The T^2 index using DIPLS is the red broken line which does not reveal the deviation from the principal manifold. The perpendicular black broken line is the proposed residual statistic.

distance form the proposed principal manifold.

Definition 3.1 *The principal manifolds of $\hat{X}_A$ and $\tilde{X}_A$ spaces are respectively defined as the subspaces spanned by their principal loading vectors as basis, representing the directions along which maximum populations of data-points in $\hat{X}_A$ and $\tilde{X}_A$ are distributed.*

According to the above definition, the principal manifolds is spanned by basis, in which the data-point might possibly vary during the normal operating condition. To derive the principal manifolds, the first step is to determine the number of time lag between the columns of both quality-related $\hat{X}_A$ and quality-unrelated $\tilde{X}_A$ measurement matrices. This can be done by the following method presented in [23].

Assume l is the time-lag between columns of $\hat{X}_A$. If $l > 0$, $\hat{X}_A$ is required to be augmented l times according to Eq. (3.2) and derive the augmented matrix $\hat{X}^l_A$. However, in the case that $l = 0$, to ensure that the minimum dimension of the principal manifold subspace of $\hat{X}^l_A$ in Eq. (3.8) is at least one, i.e. principal manifold is a one-dimensional line, $l = 1$ is selected such that $\hat{X}^l_A$ is not of full column rank. Similarly for $\tilde{X}_A$, the number of augmentation l' can be determined using the same procedure and derive the augmented matrix $\tilde{X}^{l'}_A$.

In order to find the basis of the principal manifolds that carry direction(s) with maximum population of $\hat{X}^l_A \in \mathcal{R}^{(N-h-l)\times n(h+1)(l+1)}$ and $\tilde{X}^{l'}_A \in \mathcal{R}^{(N-h-l')\times n(h+1)(l'+1)}$, it is suggested to apply SVD as follows,

$$\begin{aligned} \hat{X}^l_A &= \begin{bmatrix} \hat{U}_{\hat{X}^l_A} & \tilde{U}_{\hat{X}^l_A} \end{bmatrix} \begin{bmatrix} \hat{S}_{\hat{X}^l_A} & 0 \\ 0 & \tilde{S}_{\hat{X}^l_A} \end{bmatrix} \begin{bmatrix} \hat{V}^T_{\hat{X}^l_A} \\ \tilde{V}^T_{\hat{X}^l_A} \end{bmatrix} \\ \tilde{X}^{l'}_A &= \begin{bmatrix} \hat{U}_{\tilde{X}^{l'}_A} & \tilde{U}_{\tilde{X}^{l'}_A} \end{bmatrix} \begin{bmatrix} \hat{S}_{\tilde{X}^{l'}_A} & 0 \\ 0 & \tilde{S}_{\tilde{X}^{l'}_A} \end{bmatrix} \begin{bmatrix} \hat{V}^T_{\tilde{X}^{l'}_A} \\ \tilde{V}^T_{\tilde{X}^{l'}_A} \end{bmatrix} \end{aligned} \tag{3.8}$$

The direction(s) with maximum population distribution required for determining the principal manifolds of $\hat{X}_A^l$ and $\tilde{X}_A^{l'}$ are the columns of the $\hat{V}_{\hat{X}_A^l}$ and $\hat{V}_{\tilde{X}_A^{l'}}$ corresponding to relatively dominant singular values of the $\hat{S}_{\hat{X}_A^l} \in \mathcal{R}^{d \times d}$ and $\hat{S}_{\tilde{X}_A^{l'}} \in \mathcal{R}^{d' \times d'}$, respectively. In Eq. (3.8), d and d' are the number of dominant singular values and it can be determined by following the cumulative percentage variance (CPV) technique [72].

In order to monitor deviation from the obtained principal manifold, as can be seen in Fig. 3.2, we propose to use Q statistic which geometrically measures the *Euclidean* distance between each data-point of $\hat{X}_A^l$ and $\tilde{X}_A^{l'}$ from their corresponding principal manifolds. For this purpose, the $\tilde{V}_{\hat{X}_A^l}$ and $\tilde{V}_{\tilde{X}_A^{l'}}$ are derived using Eq. (3.8). The Q statistic for each new given point $\hat{x}_A^l \in \mathcal{R}^{n(h+1)(l+1)}$ and $\tilde{x}_A^{l'} \in \mathcal{R}^{n(h+1)(l'+1)}$ are the new proposed monitoring indices for quality output-related fault detection as follows,

$$\begin{cases} Q_{\hat{x}_A^l} & = \hat{x}_A^l \tilde{V}_{\hat{X}_A^l} \tilde{V}_{\hat{X}_A^l}^T \hat{x}_A{}^{lT} \\ Q_{\tilde{x}_A^{l'}} & = \tilde{x}_A^{l'} \tilde{V}_{\tilde{X}_A^{l'}} \tilde{V}_{\tilde{X}_A^{l'}}^T \tilde{x}_A{}^{l'T} \end{cases} \tag{3.9}$$

Fig. 3.2 shows a pictorial example of the scores' manifold for a dynamic linear time-variant process that has a three dimensional $\hat{X}_A^l$ with the column rank of $d = 2$. Then the corresponding principal manifold has two basis. The *Euclidean* distance of the decomposed process variables from the determined principal manifold is the new residual index (i.e. black broken lines perpendicular to the manifold in Fig. 3.2) which is utilized to monitor the deviation of $\hat{X}_A^l$ from their underlying vector auto-regressive (VAR) structure. Therefore, as long as the process is in the normal operation, even if the mean values of $\hat{X}_A^l$ and $\tilde{X}_A^{l'}$ change with time, they vary along the derived principal manifold and the proposed residuals $Q_{\hat{x}_A^l}$ and $Q_{\tilde{x}_A^{l'}}$ will not exceed their normal thresholds. On the other hand, as shown in Fig. 3.2 by broken red lines, the T^2 index defined in Eq. (3.7) may violate its threshold even for the normal variation of the process variables.

According to [73], the upper control limit for index $Q_{\hat{X}_A^l}$ is defined as,

$$UCL_{Q_{\hat{X}_A^l}} = \hat{\theta}_1 \left(\frac{\hat{z}_\alpha \sqrt{2\hat{\theta}_2 \hat{g}_0^2}}{\hat{\theta}_1} + 1 + \frac{\hat{\theta}_2 \hat{g}_0 (1 - \hat{g}_0)^2}{\hat{\theta}_1} \right)^2 \tag{3.10}$$

where, $\hat{\theta}_i = \sum_{j=1}^{n(h+1)(l+1)-d} \tilde{S}_{\hat{X}_A^l}(j)$, $\hat{g}_0 = 1 - \dfrac{2\hat{\theta}_1 \hat{\theta}_3}{3\hat{\theta}_2^2}$ and $\hat{z}_\alpha$ is the value of the normal distribution function with confidence level α and the degree of freedom $n(h+1)(l+1)$ and $N - n(h+1)(l+1)$. By following similar procedure, the upper control limit $UCL_{Q_{\tilde{X}_A^{l'}}}$ can be determined.

In *Algorithm 2*, the step by step procedure for the proposed PLS-based approach is summarized. The proposed PLS-based approach consists of an off-line training step and a real-time testing counterpart.

Algorithm 2 Cascade PLS approach for fault detection in non-stationary processes

```
procedure TRAINING:
    Collect normal process variables data ($X$) and quality-outputs ($Y$),
    Construct $X_A$ as shown in Eq. (3.2) and mean-center each columns of $X_A$ and $Y$,
    Calculate the correlation matrix $\psi$ by Eq. (3.4) and utilize it for determining the $\hat{X}_A$ and $\tilde{X}_A$ as shown in Eq. (3.6),
    Determine the parameter $l$ and $l'$, then construct $\hat{X}^l_A$ and $\tilde{X}^{l'}_A$.
    Conduct SVD on $\hat{X}^l_A$ and $\tilde{X}^{l'}_A$ to determine $\tilde{V}_{\hat{T}^l_A}$, $\tilde{S}_{\hat{X}^l_A}$, $\tilde{V}_{\tilde{X}^{l'}_A}$ and $\tilde{S}_{\tilde{X}^{l'}_A}$ as shown in Eq. (3.8),
    Determine $UCL_{Q_{\hat{X}^l_A}}$ and $UCL_{Q_{\tilde{X}^{l'}_A}}$ by following Eq. (3.10).
procedure TESTING:
    For a given new sample $x_{new} \in \mathcal{R}^n$, determine the mean-centered augmented $x_{new\_A} \in \mathcal{R}^{n(h+1)}$,
    By following Eq. (3.6), determine $\hat{x}_{new\_A}$ and $\tilde{x}_{new\_A}$,
    Perform augmentation on $\hat{x}_{new\_A}$ and $\tilde{x}_{new\_A}$ to build $\hat{x}^l_{new\_A}$ and $\tilde{x}^{l'}_{new\_A}$ if necessary,
    Determine the proposed new indices $Q_{\hat{x}_{A^l}}$ and $Q_{\tilde{x}^{l'}_A}$ in Eq. (3.9),
    if $Q_{\hat{x}_{A^l}} > UCL_{Q_{\hat{X}^l_A}}$ and $Q_{\tilde{x}^{l'}_A} > UCL_{Q_{\tilde{X}^{l'}_A}}$ then
        $\Rightarrow$ Quality output-related fault exists
    if $Q_{\hat{x}_{A^l}} < UCL_{Q_{\hat{X}^l_A}}$ and $Q_{\tilde{x}^{l'}_A} > UCL_{Q_{\tilde{X}^{l'}_A}}$ then
        $\Rightarrow$ Quality output-related fault exists
    else
        $\Rightarrow$ Normal operation
```

It is worth noting that the collected data-points for the training step must include certain time-varying operating conditions of the process. This time-varying mode does not need to match the actual mode in the testing step and it is only required to learn the underlying principle manifold.

3.4 Simulation Results and Case Studies

In this section, the proposed strategy is conducted on a numerical example and a non-isothermal CSTR chemical process. Moreover, the results of the proposed method are compared with the dynamic improved PLS method [1] in order to clarify the contribution of the proposed algorithm and its performance for time-variant cases. A false alarm rate (FAR) index is considered to inspect the accuracy of the proposed algorithm. The lower FAR for normal conditions represents the better performance of the method to capture the normal time-varying information of the process.

$$FAR = \frac{number\ of\ false\ alarms}{total\ numberofsamples} \times 100 \tag{3.11}$$

3.4.1 Numerical Example

A time-variant version of the numerical example studied in [1] is considered here as following,

$$\begin{aligned} U_i &= W_1U_{i-1} + W_2U_{i-2} + g_i + \mu_i \\ X_i &= BU_i + f_i + \nu_i, \\ Y_i &= C_1X_i + C_2X_{i-1} + e_i \end{aligned} \tag{3.12}$$

where, $\nu, e \sim N(0, 0.02, I_5)$ and $\mu \sim N(0, 2^2, I_2)$. f_i is the additive fault to the process variables. g_i is the additive time-varying component of the process variables and may be any function of the increment i. The following is the matrices in Eq. (3.12),

$$\begin{aligned} W_1 &= \begin{bmatrix} 0.4389 & 0.1210 & -0.0862 \\ -0.2966 & -0.0550 & 0.2274 \\ 0.4538 & -0.6573 & 0.4239 \end{bmatrix}, \\ W_2 &= \begin{bmatrix} -0.2998 & -0.1905 & -0.2669 \\ -0.0204 & -0.1585 & -0.2950 \\ 0.1461 & -0.0775 & 0.3749 \end{bmatrix}, \\ B &= \begin{bmatrix} 0.5586 & 0.2042 & 0.6370 \\ 0.2007 & 0.0492 & 0.4429 \\ 0.0874 & 0.6062 & 0.0664 \\ 0.9332 & 0.5463 & 0.3743 \\ 0.2594 & 0.0958 & 0.2491 \end{bmatrix} \\ C_1 &= \begin{bmatrix} 0.9249 & 0.4350 \\ 0.6295 & 0.9811 \\ 0.8783 & 0.0960 \\ 0.6417 & 0.5275 \\ 0.7984 & 0.5456 \end{bmatrix} C_1 = \begin{bmatrix} 1.7198 & -0.3715 \\ 0.5835 & 1.5011 \\ 1.4236 & 1.3226 \\ 0.4963 & -1.4145 \\ -2.5717 & 1.0696 \end{bmatrix} \end{aligned} \tag{3.13}$$

For off-line training step, 10,000 samples are generated with sample time $T = 0.01$ sec and 20,000 samples are collected for real-time analysis. In this numerical example, the process variables utilized for training step have a ramp trend as $g_i = 0.03 \times (i-1)T,\ i = 1, ..., 10,000$. However, after sample $10,000^{th}$, their time-varying trend become a random drift as $g_i = 0.03 \times (i-1)T + 0.015 \times D(i),\ i = 10,000, ..., 20,000$, where, $D(i)$ is the added random drift. This choice of g_i is to demonstrate that the time-varying trend of process variables do not need to be similar for both training and real-time analysis. In this simulation, we compare the proposed PLS-based approach with the Improved PLS method [1] to demonstrate that the proposed method achieved the non-stationary fault detection with the same calculation complexity of the IPLS method. It should be noted that the application of adaptive PLS is also a fair comparison in this chapter, but since both methods will lead to the approximately same false alarm rate (FAR) result, it may not clearly indicate the contribution of the proposed PLS-based approach.

The quality output-related additive fault vector is considered $f^1 = [2, 1, -3, 2, -5]^T$. The fault vector $f^2 = [0.0054, 0.3145, -0.0432, 0.7516, -0.4440]^T$ which is perpendicular to the

Table 3.1: FAR of the proposed approach and DIPLS method [1] for numerical example and CSTR case study.

Scenarios	Numerical Example		CSTR	
	DIPLS	New method	DIPLS	New method
No Fault	59%,21%	9%,7%	46%,5%	1%,2%
Fault 1	39%,22%	4%,8%	18%,18%	2%,4%
Fault 2	24%,19%	8%,2%	31%,8%	1%,7%

subspace of Y is also considered as the quality output-unrelated fault vector, thus, it does not affect output Y in Eq. (3.12).

Fig. 3.3 shows the process variables and outputs for the normal operation. In Figs. 3.3(a) and 3.3(d), the T^2 index of the DIPLS method [1] gives false alarm in the normal mode because of the non-stationarity in the process time-series. Fig. 3.5 shows the results when a quality output-related malfunction f^1 is introduced to the process. As can be seen in Fig. 3.5(a), T^2 quality output-related index crosses the UCL around $10,000^{th}$ sample, at which the process is still in the normal operation. However, both $Q_{\hat{X}^l_A}$ and $Q_{\tilde{X}^l_A}$ can successfully detect the quality output-related malfunction, (see Figs. 3.5(c) and 3.5(d)).

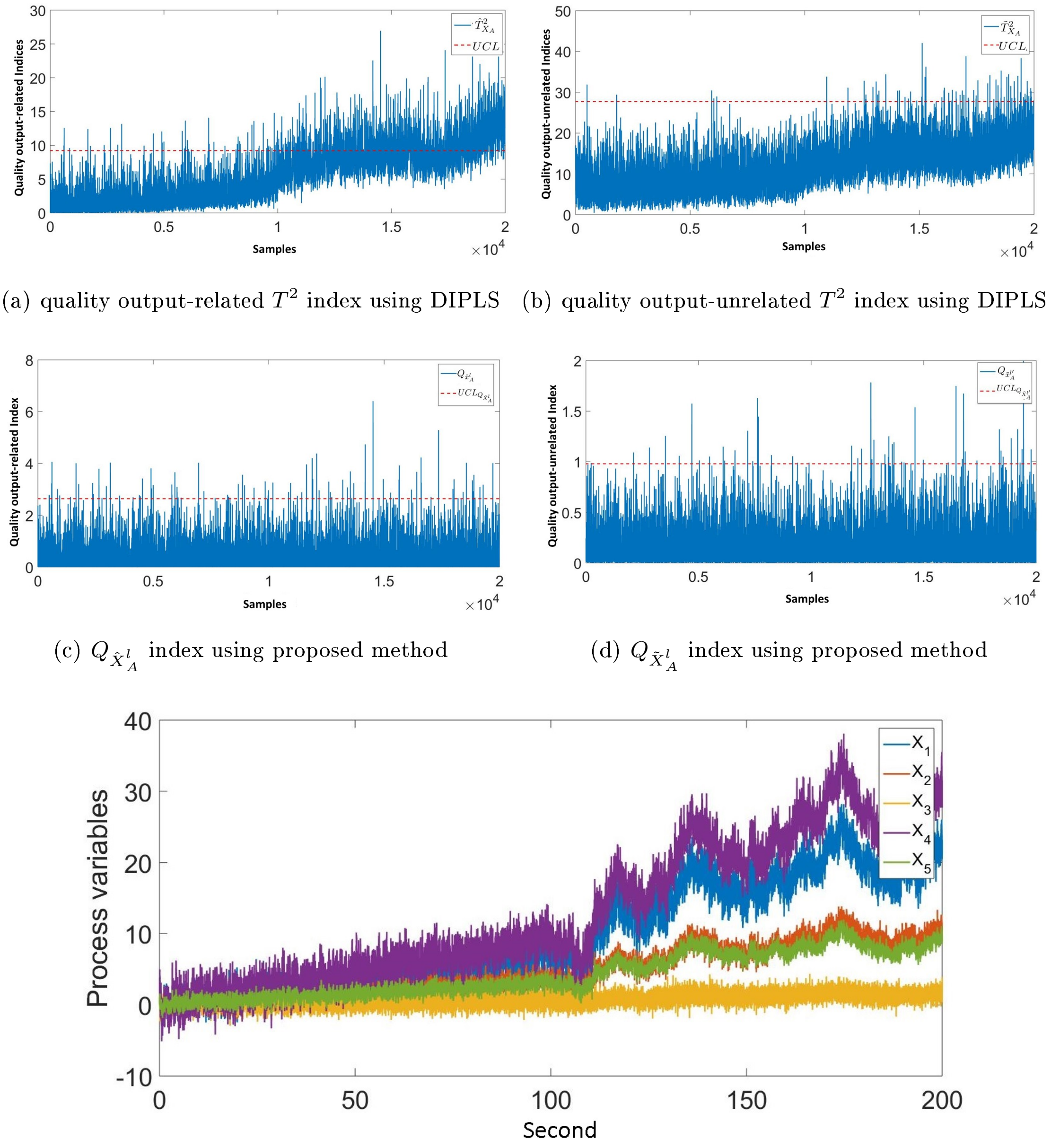
As shown in Figs. 3.6(c) and 3.6(d), the fault f^2 which does not affect the quality output is also successfully detected by using the proposed PLS-based approach . However, quality output-related T^2 index in Fig. 3.6(a) produces false alarms while the process quality output remains intact. The FAR for both methods and two operating scenarios are determined and presented in Table 3.1, which indicates a significant improvement in reducing FAR.

3.4.2 Case Study on Continuous Stirred Tank Reactor (CSTR)

In this section, the first-order non-isothermal CSTR reactor is under study. This process is utilized as a viable case study in [74] [19] [48] for fault detection and diagnosis purposes due to the time-varying nature of its process variables. Fig. 3.4 is the schematic diagram of the CSTR reactor indicating the inputs, intermediate measurements, and quality-outputs.

In this process ($A \rightarrow B$), the reactant A is premixed with the solvent flow in order to produce B. There is one feed stream into the reactor including the reactant (A) mixed with a solvent. Also, there is a cooling water stream that regulates the temperature of the process. As can be seen in Fig. 3.4, the reactant and cooling water flows F_A and F_c control the output concentration C_A and temperature, respectively. Also, in this process, there exists a PI controller that regulates the temperature.

By assuming the ideal mixture and constant physical properties of the materials, the material and energy balance equations of the CSTR process is as follows,



(a) quality output-related T^2 index using DIPLS

(b) quality output-unrelated T^2 index using DIPLS

(c) $Q_{\hat{X}^l_A}$ index using proposed method

(d) $Q_{\tilde{X}^l_A}$ index using proposed method

(e) Process variables in normal operating condition

Figure 3.3: Simulation results of the numerical example for normal mode (no-fault) using the proposed method and DIPLS.

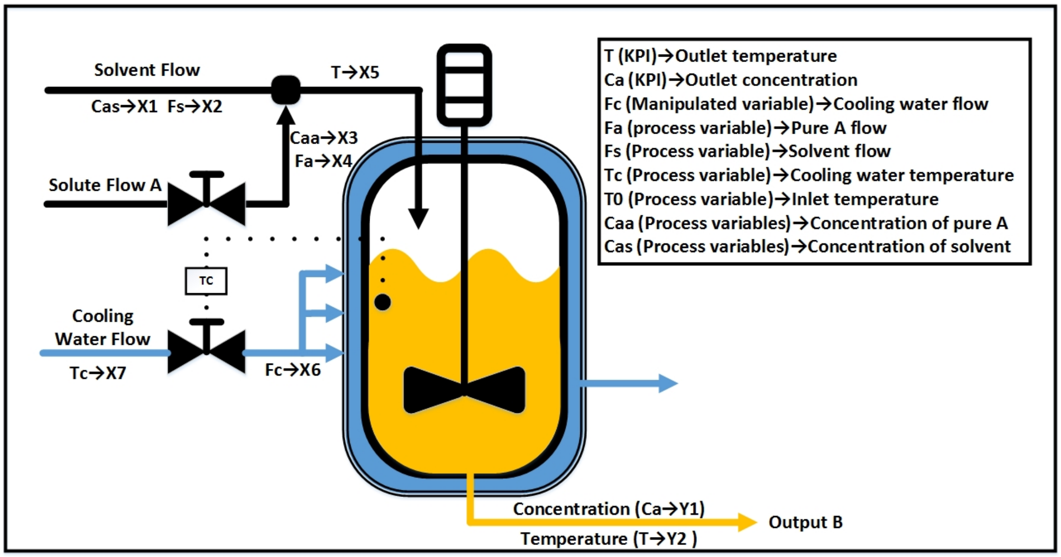
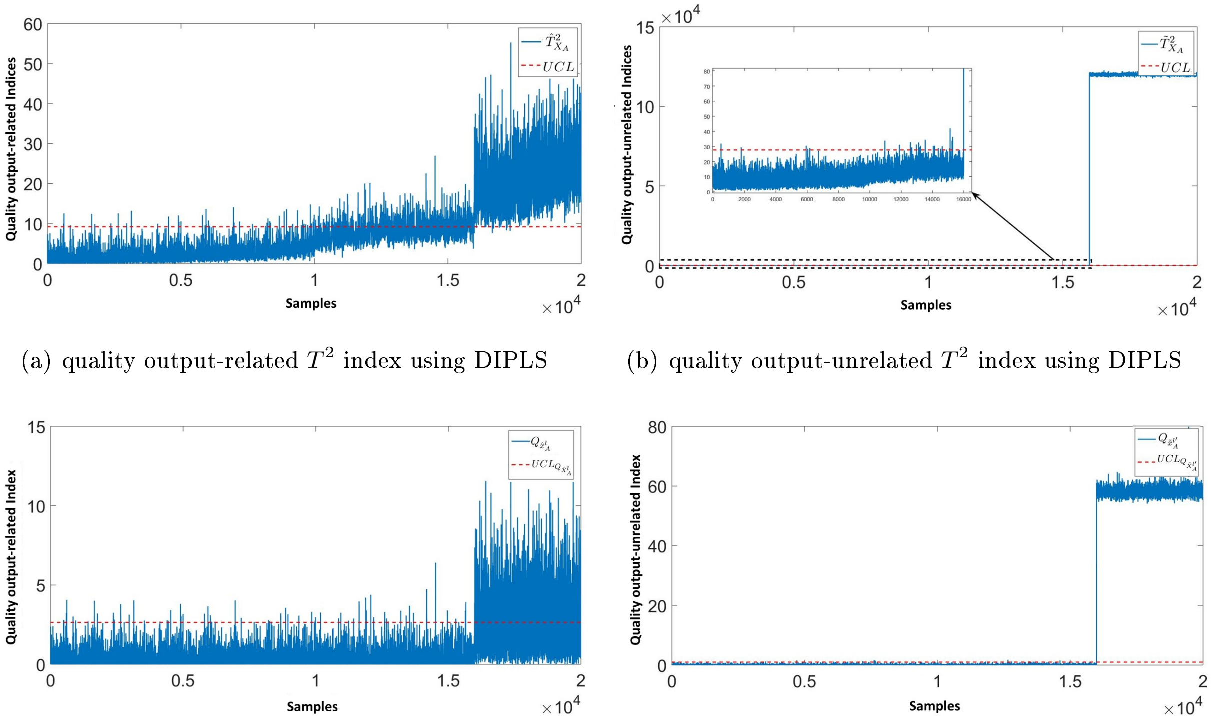

Solvent Flow
T→X5
Cas→X1 Fs→X2
Caa→X3
Fa→X4
Solute Flow A
TC
Cooling
Water Flow
Tc→X7
Fc→X6
T (KPI)→Outlet temperature
Ca (KPI)→Outlet concentration
Fc (Manipulated variable)→Cooling water flow
Fa (process variable)→Pure A flow
Fs (Process variable)→Solvent flow
Tc (Process variable)→Cooling water temperature
T0 (Process variable)→Inlet temperature
Caa (Process variables)→Concentration of pure A
Cas (Process variables)→Concentration of solvent
Concentration (Ca→Y1)
Temperature (T→Y2)
Output B

Figure 3.4: Non-isothermal continuous stirred tank reactor (CSTR) process.



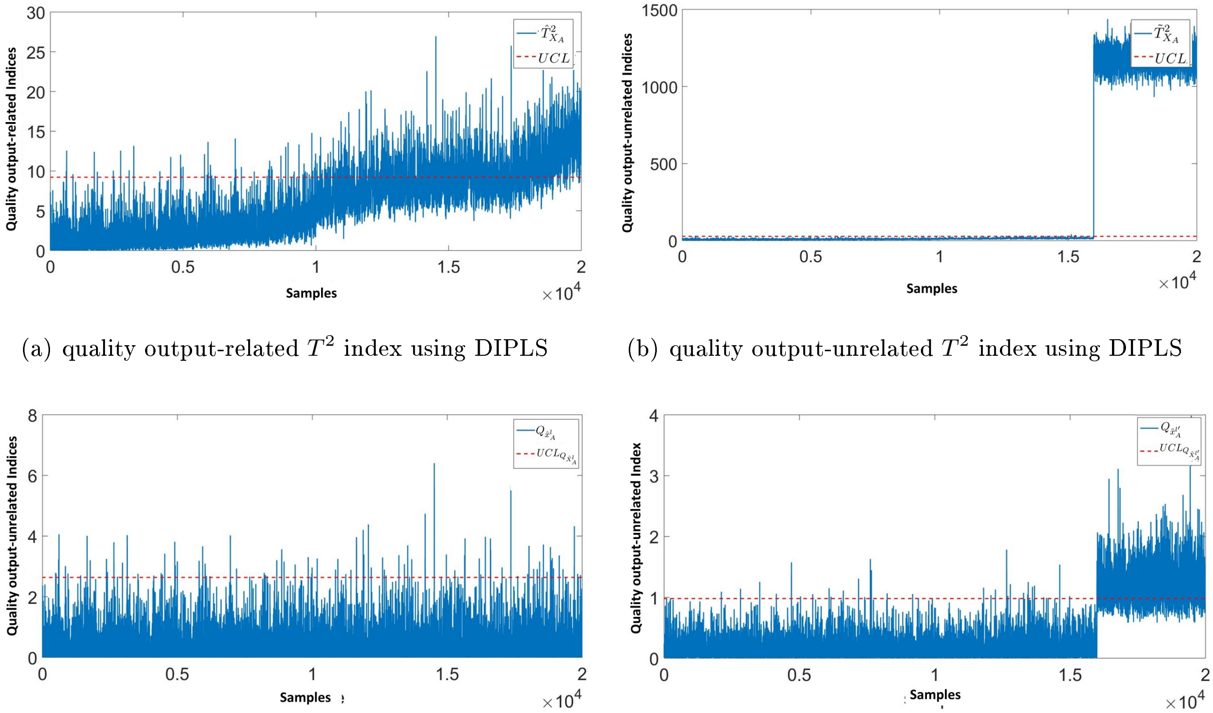
(a) quality output-related T^2 index using DIPLS

(b) quality output-unrelated T^2 index using DIPLS

(c) $Q_{\hat{X}^l_A}$ index using proposed method

(d) $Q_{\tilde{X}^l_A}$ index using proposed method

Figure 3.5: Simulation result of numerical example for quality output-related fault f^1 using the proposed method and DIPLS.



(a) quality output-related T^2 index using DIPLS

(b) quality output-unrelated T^2 index using DIPLS

(c) quality output-related $Q_{\hat{X}^l_A}$ index using proposed method

(d) quality output-unrelated $Q_{\tilde{X}^l_A}$ index using proposed method

Figure 3.6: Simulation result of numerical example for quality output-unrelated fault f^2 using the proposed method and DIPLS.

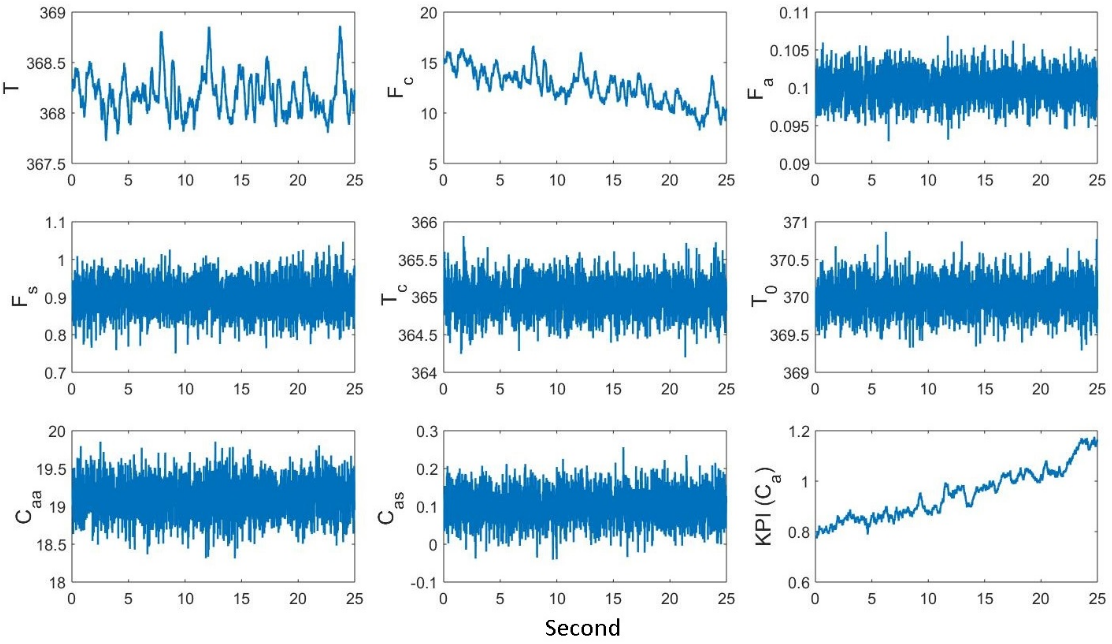


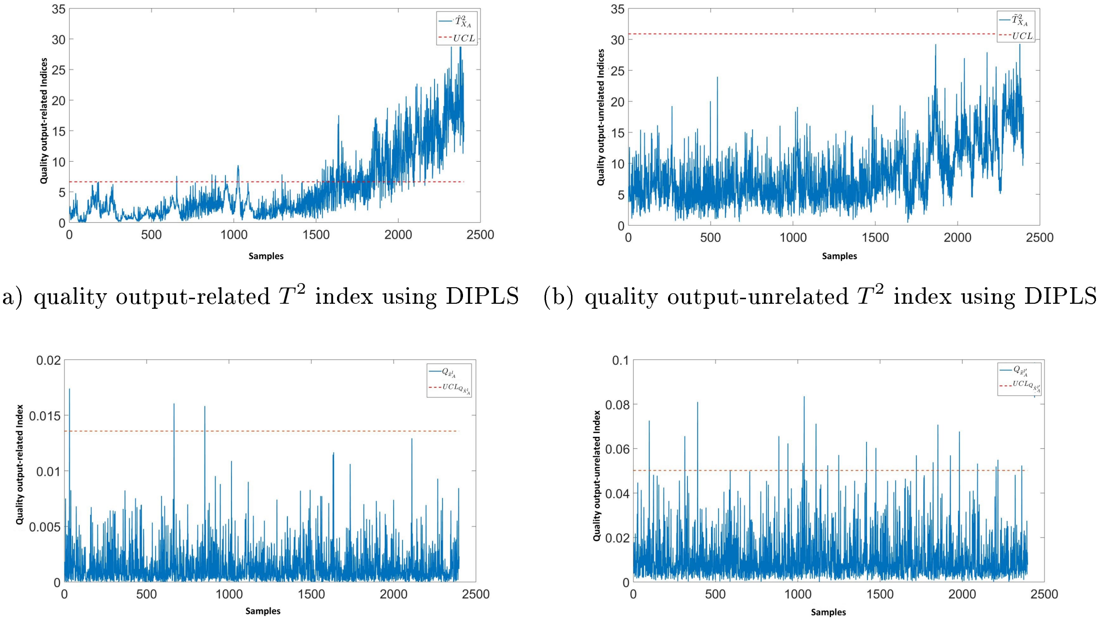
Figure 3.7: The CSTR process measurements in normal operating condition

$$
\begin{aligned}
\frac{dC_a}{dt} &= \frac{F}{V}\frac{C_{aa}F_a + C_{as}F_s}{F_a + F_s} - \frac{F}{V}C_a - k_0 e^{(-\frac{E}{RI})}C_a \\
V_\rho C_p \frac{dT}{dt} &= \rho C_p F(T_0 - T) - \frac{a2aF_c^{b+1}}{F_c + a_2 a F_c^b/2\rho_c C_{pc}}(T - T_c) + (-\Delta H)V_{a_1} k_0 e^{(-\frac{E}{RI})}C_a
\end{aligned}
\tag{3.14}
$$

where the empirical relationship between the heat transfer coefficient and the flow of the cooling water is assumed $UA = aF_c^b$. The process disturbances is introduced by multiplication of a random coefficient to the reaction constant $K = a_1 k_0 e^{(\frac{E}{RT})}$ and heat transfer coefficient $UA = a_2 a F_c^b$. The additive disturbances to the process are in the form of an auto-regressive and a random independent noise. The list of the process variable is shown in the Fig. 3.4. The time-variation of the process variables is due to the deactivation of the catalyst and it is simulated through adjustment of pre-exponential factor $K_0 = (1 - t \times 10^{-4})K_0^{initial}$. More detailed information about the coefficients and PI controller used in this simulation can be found in [19]. The process is simulated for 2,500 minutes with a sample time of 1 minute, which 750 samples are chosen for training.

Fig. 3.7 shows the process time-series for the normal operating condition. As can be concluded from the Fig. 3.8, both the quality output-related and quality output-unrelated T^2 indices using DIPLS violate their UCL for the normal operating mode. However, $Q_{\hat{X}_A}$ and $Q_{\tilde{X}_A}$ are not sensitive to time-variation of process variables and remain below their corresponding UCLs since they operate along with their ARMA model without the presence of faults.

The process is subjected to two types of faults as following,



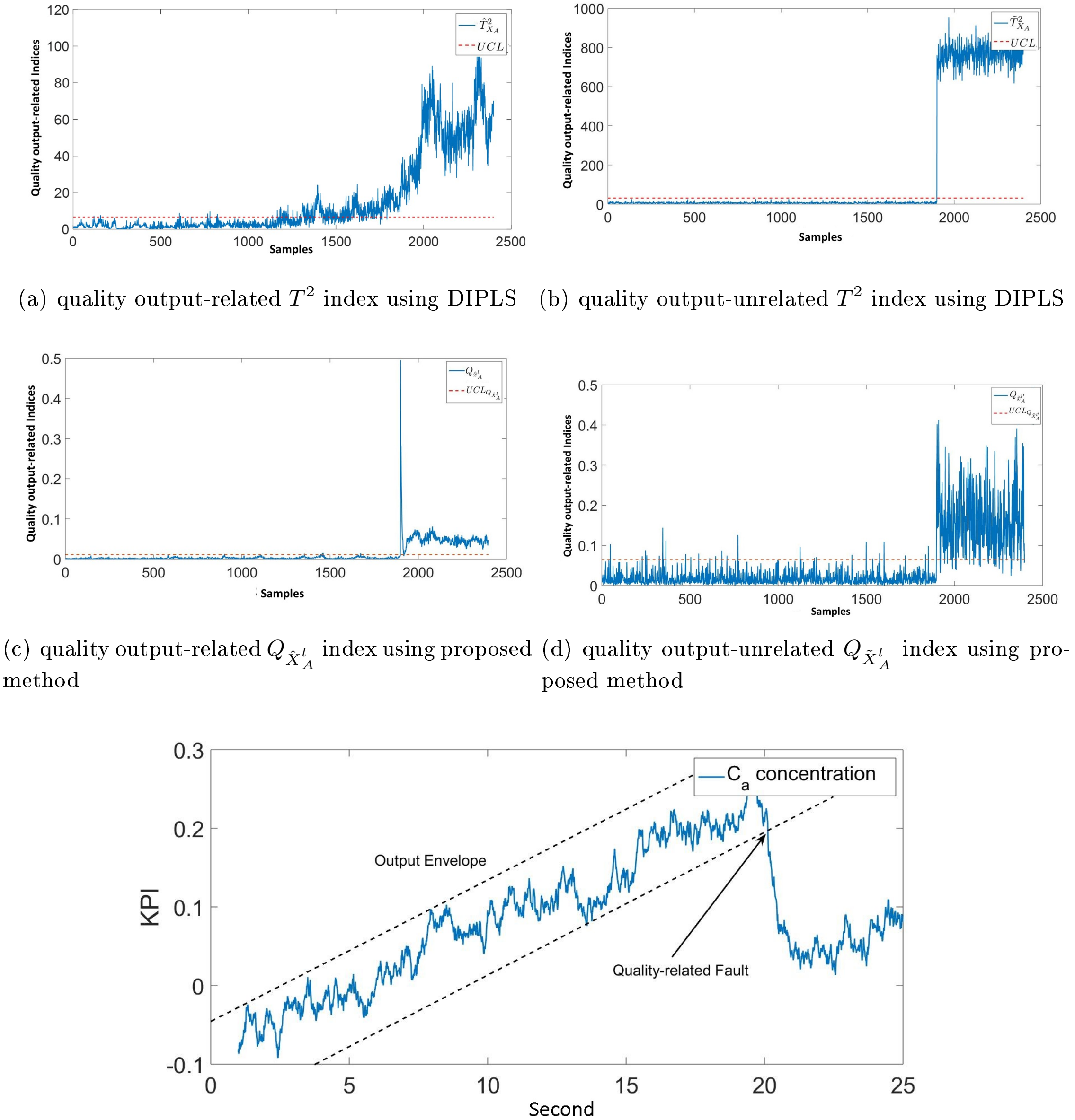
(a) quality output-related T^2 index using DIPLS

(b) quality output-unrelated T^2 index using DIPLS

(c) quality output-related $Q_{\hat{X}^l_A}$ index using proposed method

(d) quality output-unrelated $Q_{\tilde{X}^l_A}$ index using proposed method

Figure 3.8: Simulation result of CSTR benchmark for normal operating condition comparing DIPLS with the proposed scheme.



(a) quality output-related T^2 index using DIPLS

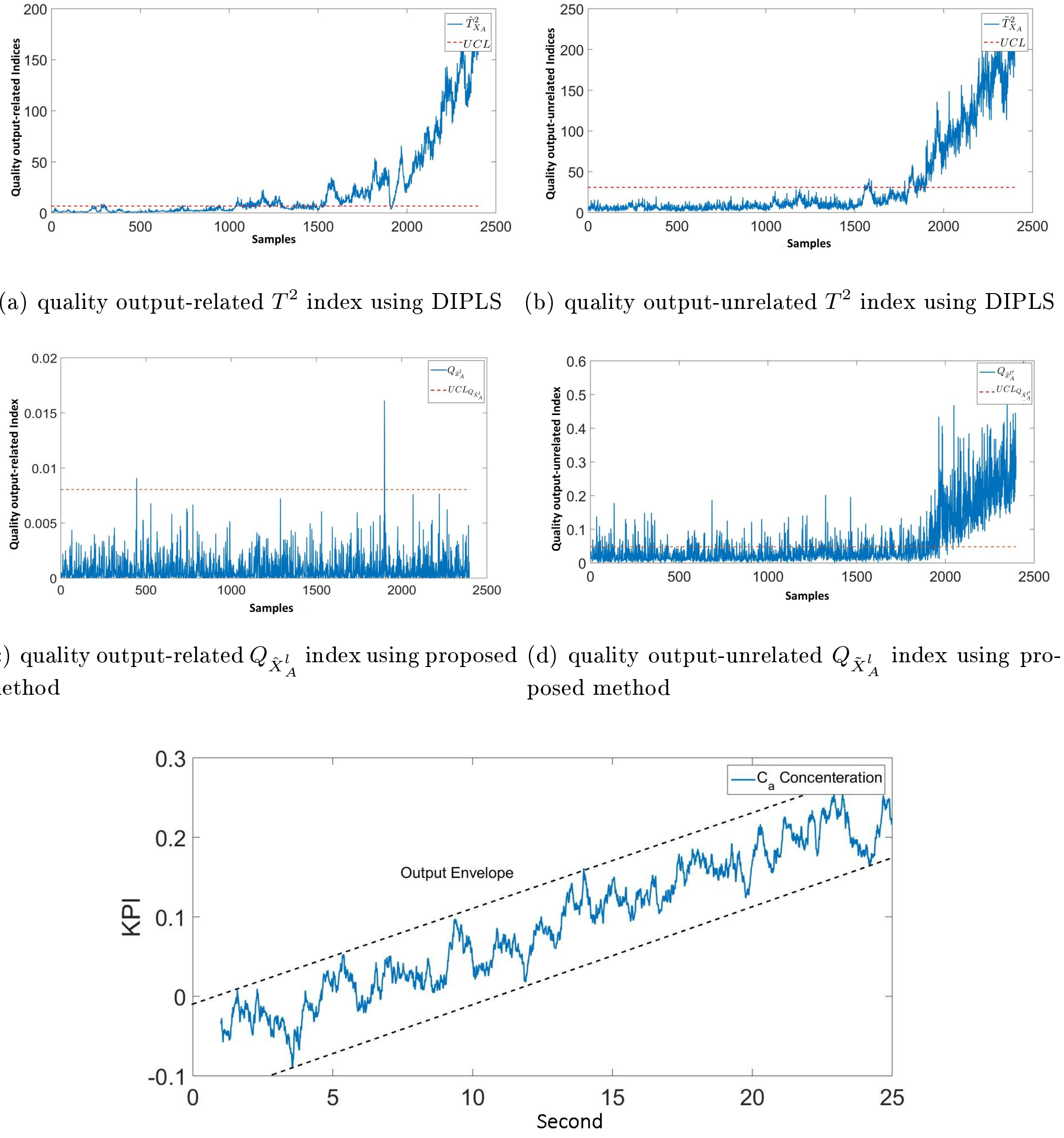
(b) quality output-unrelated T^2 index using DIPLS

(c) quality output-related $Q_{\hat{X}^l_A}$ index using proposed method

(d) quality output-unrelated $Q_{\tilde{X}^l_A}$ index using proposed method

(e) C_a concentration for the first fault scenario

Figure 3.9: Simulation result of CSTR case study for *Fault 1* scenario.



(a) quality output-related T^2 index using DIPLS

(b) quality output-unrelated T^2 index using DIPLS

(c) quality output-related $Q_{\hat{X}^l_A}$ index using proposed method

(d) quality output-unrelated $Q_{\tilde{X}^l_A}$ index using proposed method

(e) C_a concentration for the second fault scenario

Figure 3.10: Simulation result of CSTR case study for *Fault 2* scenario.

Fault 1: A step malfunction of 3.0 C° in inlet cooling water temperature T_c starting at $2,000^{th}$ sample,

Fault 2: A step malfunction of 5.0 C°in the inlet temperature T_0 starting at $2,000^{th}$ sample.

Fig. 3.9(e) shows that for *Fault 1* scenario, the quality-output C_a is deviated from its expected envelop, thus, subjected to a fault. Therefore, this fault scenario is a quality output-related case. Figs. 3.9(a) to 3.9(d) show that the proposed $Q_{\hat{X}_A^l}$ and $Q_{\tilde{X}_A^l}$ indices successfully detected the quality output-related fault. However, the T^2 index used in DIPLS method (see Fig. 3.9(a) and 3.9(b)) produces false alarm before the fault truly happens because it miss detects the normal mean changes as a fault.

In *Fault 2*, the 5 C^0 step change in the inlet temperature is not significant enough and it is compensated by the incorporated PI controller, thus, as indicated in Fig. 3.10(e), the quality output is not affected by this malfunction. Therefore, this scenario is deemed as a quality output-unrelated case. Fig. 3.10(c) shows that the $Q_{\hat{X}_A}$ remains below its UCL while $Q_{\tilde{X}_A}$ in Fig. 3.10(d) shows the present of a malfunction in the $\tilde{X}_A$. Hence, by utilizing the proposed method, the quality output-unrelated fault is also successfully detected. Even though the T^2 quality output-unrelated index is showing the presence of a fault in Fig. 3.10(b), the quality output-related T^2 index falsely indicates presence of a quality output-related fault. The comparative FAR for both fault scenarios is presented in Table 3.1. The results show that the proposed method can successfully compartmentalize the time-varying mean changes of measurements under normal condition form quality output-related/unrelated faults. In general, as can be concluded from the result of the DIPLS method as the most recent improved version of the PLS-based approach, it performs poorly for processes subjected to non-stationary variations and has a high FAR.

3.5 Summary

In this study, a modified PLS-based framework is proposed to distinguish quality output-related process faults from the time-variation of measured variables in the normal operating mode. In contrast with the state-of-the-art methods, such as DIPLS, which measures the distance of each data-point from the origin and consequently fails to capture the normal time-varying operating condition, the proposed scheme forms the monitoring model built upon the new concept of principal manifolds using the time-varying information of process variables. Therefore, the proposed method can successfully distinguish normal time-varying changes from the quality output-related and unrelated faults in dynamic linear processes. It should be noted that the proposed method assumes the normal non-stationary mean-changes of the process should be seen in the training data which will lead to preserving the ARMA structure of each component in the principal manifold. Moreover, the suggested solution assumes that the process base-line model stays unchanged during the normal non-stationary variations. Finally, two case studies on a numerical example as well as a CSTR benchmark are conducted to demonstrate the efficiency of this method.

Chapter 4

Root-Cause Fault Diagnosis Using Symbolic Dynamic Transfer Entropy

4.1 Introduction

Fault diagnosis and root cause identification of an abnormal event are considered as complex and time-consuming tasks. The benefits of these monitoring steps in industrial processes are presented in [75] and some of the improvements in fault diagnosis are reviewed. For conducting data-driven fault diagnosis, causality analysis is a viable solution. By using this approach, the causal connections between process measurements are investigated. Since the symptoms of the process fault propagate to all measurement time-series through these causal connections, i.e. smearing out effect, knowledge of these connections can be effectively utilized to diagnose the root-cause fault. Various techniques have been used for this purpose such as transfer entropy (TE) [57] [76] [58]; Granger causality (GC) [54] [77] [63], cross-correlation with time lag [78], partial directed coherence [79], and convergent cross-mapping [80].

Selecting the proper method for conducting causality analysis always depends on statistic nature of time-series and the application. Among all available approaches, GC and TE have gained relatively more interests due to their simple structure and their compatibility with industrial processes. For these two well-known approaches, Lindner *et al* [81] provided a comparative analysis, and it can be used to decide which method is a proper choice given a particular application. These two techniques have been proven effective for root-cause fault diagnosis in industrial processes as shown in some research investigations. In this chapter, TE is selected as the main tool for conducting causality analysis since it can be applied to both linear and non-linear processes.

One of the conventional ways for estimating the TE between two time-series is using the kernel density estimators (KDE), which can fit a proper distribution function to the joint and conditional PDFs [59]. However, this approach has a high-computational complexity making the real-time application of TE for causality analysis an arduous task. To address this limitation, in this chapter an alternative method for estimating TE between two time-series is proposed and it is further used for root-cause fault diagnosis.

Symbolic time-series analysis (STSA) is a powerful tool for modeling and characterization of the non-linear dynamical systems [82] [83]. Following this theory, Ray in [84] proposed symbolic dynamic filtering (SDF) as a viable tool for fault detection and pattern recognition. A thorough comparison between SDF and other data-driven approaches such as PCA and artificial neural networks (ANN) is conducted in [85]. SDF has been recently proposed as a relatively fast feature extraction tool from time-series [84]. In the SDF approach, the time-series are symbolized according to a proper partitioning which is performed according to nominal time-series. It should be noted that the partitioning remains invariant during the analysis, hence, the symbolic sequences will change while the dynamical behavior of the time-series changes. Probabilistic finite state automata (PFSA) [86] is generated from the symbolic sequence of the time-series to model the dynamical behavior. Moreover, the probability distributions that are obtained from the PFSA provides a statistical representation of the behavioral pattern that can be further utilized for anomaly detection and diagnosis.

One of the prominent advantages of the symbolic analysis of time-series data is its enhanced computational efficiency. Also, analysis of the symbolic data is often less sensitive to measurement noise and therefore its application attracts great interest while the computational efficiency is crucial. This prominent feature motivated the authors to utilize SDF formulation for efficient and fast calculation of causal dependency through TE between a pair of time-series data, which is further applied for real-time root-cause fault diagnosis. Although the application of SDF itself has been reported earlier [87] [88] [89] [90], important preliminary concepts of this technique are reviewed for the completeness of the proposed framework. The proposed solution for the problem of root-cause fault diagnosis using transfer entropy in a real-time manner is presented in the following.

4.2 Proposed Framework

In this chapter, a new approach based on symbolic dynamic filtering is proposed, which can be applied as a fast alternative for estimating transfer entropy between two times series for causality analysis. Furthermore, a general framework is proposed for conducting root-cause fault diagnosis by using the proposed approach which reveals the root-cause variable(s) that holds responsible to be the source of the detected fault. Symbolic dynamic filtering (SDF) is used as an alternative for conventional kernel density estimators (KDEs) to estimate the joint and conditional *Shannon* entropies in the definition of the TE. Then, by utilizing symbolic dynamic filtering-based modelling, a novel and fast procedure for estimation of conditional entropy $H(\psi_{i+h}|\psi_i^{l_1}, x_i^{l_2})$ in Eq. (4.4) is proposed. To achieve this, a new concept of joint *xD-Markov* machine is introduced (See Def. 4.3).

The schematic diagram that summarizes the main steps of the proposed root-cause fault diagnosis strategy is shown in Fig. 4.1. In the first part, a reduced-rank kernel trick introduced in [34] is applied to project the process variables $X \in \mathcal{R}^{N\times m}$ with non-linear relations onto a higher dimensional linear space $\Phi(X) \in \mathcal{R}^{N\times f}$. Then, a residual index that represents the

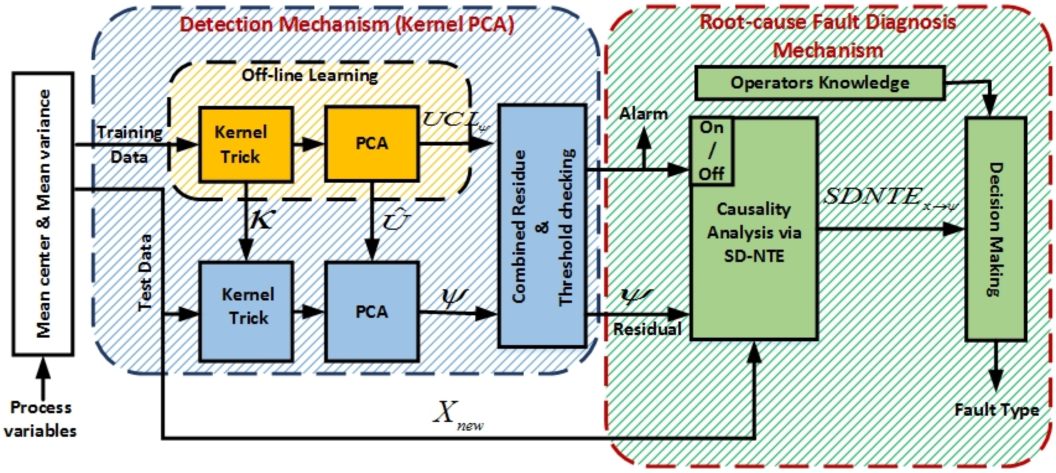


Figure 4.1: Schematic diagram of the proposed root-cause fault diagnosis algorithm.

existing fault in the process is calculated. In the second part (root-cause diagnosis mechanism in Fig. 4.1), it is proposed to find the strength of causality (measured by the proposed symbolic dynamic-based normalized transfer entropy (SDNTE)) from each process $x \in X$ variables to the residual signal ψ. The underlying idea behind the proposed framework is that the source of the fault has a stronger causal contribution to the residual signal, while in the normal (i.e. fault free) situation, there is no significant causal pathway between process variables and the residual signal. It should be noted that the relationship between the generated residual signal ψ and process variables X may be non-linear, thus, normalized transfer entropy (NTE) which is applicable to non-linear relations is utilized for causality analysis.

4.3 Residual Generation Using Kernel Trick

Assume that the given non-linear process contains m process variables $x(i) = [x_1(i), x_2(i), ..., x_m(i)] \in \mathcal{R}^m$. In an off-line measurement, $N >> m$ samples of process variables are observed and put into a matrix $X \in \mathcal{R}^{N\times m}$ for training purposes. Later, $x^* \in \mathcal{R}^{1\times m}$ indicates a single test measurement vector. The common way that has been used for dealing with non-linearity of the process variables is to map them into a high-dimensional feature subspace where the mapped version of variables follow approximately a linear dependency [29] [33]. For this purpose, consider Φ to be a non-linear function that maps $x(i) \in \mathcal{R}^{1\times m}$ into a new feature subspace $\Phi(x(i)) = [\phi(x_1(i)), \phi(x_2(i)), ..., \phi(x_m(i))] \in \mathcal{R}^{1\times f}$, $f >> m$, where in theory, f may tend to infinity. By assuming that the mapped training matrix $\Phi(X)$ is already mean centered, the covariance matrix in the feature space is determined as $C^\Phi = \dfrac{1}{N}\sum_{j=1}^{N} \Phi(x(j))\Phi(x(j))^T$. Instead of carrying out the map Φ and directly eigen-decomposing C^Φ, *kernel trick* can be alternatively used for simplifying dot product in the above high dimensional feature space without explicit knowledge of non-linear mapping function $\Phi(X)$ [31]. To this aim, dot product can be replaced with different types of kernels such as a

$N \times N$ Gram kernel matrix $K(x(i), x(j)) =< \phi(x(i)), \phi(x(j)) >$. Therefore, for the training data-set $X \in \mathcal{R}^{N\times m}$, we can find the training kernel matrix $\mathcal{K} = \Phi(X)^T\Phi(X) \in \mathcal{R}^{N\times N}$.

Since the necessary condition for the training data $\Phi(X)$ is to be mean-centered, the kernel training matrix must be centralized. Assuming $\mathcal{K}$ is obtained from uncentered training data, the mean-centered kernel K can be obtained as following,

$$K = (I_N - E_N)\mathcal{K}(I_N - E_N) \ \in \mathcal{R}^{N\times N} \tag{4.1}$$

where, I_N is $N \times N$ identity matrix and $E_N = \dfrac{1}{N}1_N 1_N^T$ such that $1_N = [1, ..., 1]^T$. Later, once a single test measurement vector $x^* \in \mathcal{R}^{1\times m}$ is available, its uncentered kernel vector $\kappa(x^*) = [\kappa(x(1), x^*), ..., \kappa(x(N), x^*)]^T \in \mathcal{R}^{N\times 1}$ can be mean centered as follows,

$$k(x^*) = (I_N - E_N)\left[\kappa(x^*) - \frac{1}{N}\mathcal{K}1_N\right] \ \in \mathcal{R}^{N\times 1}. \tag{4.2}$$

In [34], it is shown that the effective rank of training kernel matrix K is $r \leq N-1$. Hence, instead of eigen-decomposing K itself that is defined in the mapped space $\Phi(X)$, a reduced-dimensional subspace of the feature space $\Phi(X)$ so-called P can be obtained by conducting singular value decomposition as follows,

$$K = \begin{bmatrix} U_k \ \star \end{bmatrix} \begin{bmatrix} \Lambda_k \ 0 \\ 0 \ \star \end{bmatrix} \begin{bmatrix} U_k^T \\ \star \end{bmatrix} \tag{4.3}$$

where, $U_k \in \mathcal{R}^{N\times r}$ and $\Lambda_k = diag(\lambda_1, ..., \lambda_r) \in \mathcal{R}^{r\times r}$ are the corresponding non-zero eigenvectors and eigenvalues of K, respectively. It can be proven that the columns of $\Pi = \Phi(X)U_k\Lambda_k^{-1/2} = [\pi_1, ..., \pi_r] \in \mathcal{R}^{f\times r}$ contains the orthonormal bases of the reduced-rank subspace P. By considering $\Phi_P(X)$ as the projection of $\Phi(X)$ onto P whose bases are Π, the corresponding Cartesian coordinate is $\Phi_P(x^*) = \Pi\Lambda_k^{-1/2}U_k^T k(x^*)$, where, $k(x^*) = \Phi(X)^T\phi(x^*) \in \mathcal{R}^{1\times N}$ is the mean-centered kernel vector obtained by Eq. (4.2). Intuitively, $\Phi_P(x^*)$ can be deemed as a non-linear mapping of N-dimensional subspace onto a r-dimensional counterpart spanned by Π. Therefore, the coordinates of feature mapped training data $\Phi(X)$ is $Y = \Lambda_k^{-1/2}U_k^T K = \Lambda_k^{-1/2}U_k^T U_k \Lambda_k U_k^T = \Lambda_k^{1/2}U_k^T$.
In [34], Kwak proposed to extract the principal components of Y instead of kernel matrix K, which reduces the dimensionality while keeping the key information of the process variables. When $Y \in \mathcal{R}^{N\times r}$ matrix is obtained from the N observation of the training data, SVD can be deployed on the covariance matrix $C^Y = \dfrac{1}{N-1}Y^TY \in \mathcal{R}^{r\times r}$ to extract the principal directions with maximum distribution of r-dimensional data Y as $C^Y = \hat{U}\hat{\Lambda}\hat{U}^T$, where, $\hat{U}$ and $\hat{\Lambda}$ represent the loading vectors corresponding to non-zeros singular values $\hat{\Lambda}$. For a new test data $x^* \in \mathcal{R}^m$, $y = \Lambda^{-1/2}U^T k(x^*) \in \mathcal{R}^r$ is calculated and the square prediction error signal, $\psi = (I_r - \hat{U}\hat{U}^T)y$ is suggested to be utilize for fault detection and further in the proposed root-cause diagnosis methodology. The upper control limit for the ψ can be determined by following procedure proposed in [91]. The summary of the required steps for residual signal generation using kernel trick is explained in *Appendix* 8.

4.4 Root Cause Fault Diagnosis

4.4.1 Causality Analysis Using Transfer Entropy (TE)

In order to find the causality inference among a group of variables, this chapter studies the use of transfer entropy (TE) as a tool to measure the information flow. Assume that time series $x_a \in \mathcal{R}^N$ is independent and $x_b \in \mathcal{R}^N$ is dependent, then Eq. (4.4) is the generic definition of TE from x_a to x_b represented by *Shannon Entropy function* H,

$$\begin{aligned} TE_{x_a \longrightarrow x_b} &= H(x_b^{i+h\tau}|x_b^{l_1}) - H(x_b^{i+h\tau}|x_b^{l_1}, x_a^{l_2}) \\ &= \sum_{i=1}^{N} p(x_b^{i+h\tau}, x_b^{l_1}, x_a^{l_2}) \log \frac{p(x_b^{i+h\tau}|x_b^{l_1}, x_a^{l_2})}{p(x_b^{i+h\tau}|x_b^{l_1})} \end{aligned} \tag{4.4}$$

where $x_a^{l_2} = [x_a^i, x_a^{i-\tau}, ..., x_a^{i-(l_2-1)\tau}]$ and $x_b^{l_1} = [x_b^i, x_b^{i-\tau}, ..., x_b^{i-(l_1-1)\tau}]$ are the embedding vectors including the current and past values of the corresponding time series. h is the shifting index and τ is a scaling factor that allows the scaling in time of the embedding vectors, which can be set $\tau = h \leq 4$ as a rule of thumb [57].

$H(x_b^{i+h\tau}|x_b^{l_1})$ and $H(x_b^{i+h\tau}|x_b^{l_1}, x_a^{l_2})$ presented in Eq. (4.5) are the non-negative discrete *Shannon* conditional entropies determined for a set of time series with lengths of N samples, i.e. N is the number of samples exist in a window batch for each variable required for calculating causality.

$$\begin{aligned} H({x_b}^{i+h\tau}|{x_b}^{l_1}) &= -\sum_{i=1}^{N} p({x_b}^{i+h\tau}, {x_b}^{l_1}) \log p({x_b}^{i+h\tau}|{x_b}^{l_1}), \\ H({x_b}^{i+h\tau}|{x_b}^{l_1}, {x_a}^{l_2}) &= -\sum_{i=1}^{N} p({x_b}^{i+h\tau}, {x_b}^{l_1}, x_a^{l_2}) \log p({x_b}^{i+h\tau}|{x_b}^{l_1}, x_a^{l_2}) \end{aligned} \tag{4.5}$$

Eq. (4.4) is non-negative and its relative small value indicates no information flow from x_a to x_b. The intuition behind Eq. (4.4) is to calculate the improvement in prediction of x_b by having past information on both x_a to x_b versus the case when only information on past values of x_b is available. In order to determine TE between two time-series presented in Eq. (4.4), one can determine the conditional entropies $H({x_b}^{i+h\tau}|{x_b}^{l_1})$ and $H({x_b}^{i+h\tau}|{x_b}^{l_1}, {x_a}^{l_2})$ by estimating their joint probability density functions (PDFs). In [61], a survey is conducted on different methods for estimating PDFs using time-series data. *Kernel* functions are widely used for estimating PDFs, e.g. [56], [92] and [93]. For further details about application of *kernel* functions regarding estimation of PDFs, readers are refereed to [57] and [59]. It should be mentioned that even though *kernel* functions have been adopted in several studies, its high complexity and burdensome computation are still considered as barriers for real-time application of TE for causality analysis purposes. Therefore, this is a motivation to propose a symbolic dynamic-based approach to estimation conditional Shannon entropies shown in Eq.

(4.5) [20].

4.4.2 Symbolic Dynamic Filtering Modelling of Time-Series

In order to estimate the TE shown in Eq. (4.4) from x_a to x_b, one would use conventional kernel density estimators (KDEs) [59]. However, in this thesis, it is proposed to adopt a modeling technique known as symbolic dynamic filtering (SDF). This approach is a powerful tool for feature extraction and time series analysis. Symbolic dynamic filtering (SDF) has been recently proposed as a feature extraction tool from time-series.

In order to calculate TE between a pair of time-series, one of the time-series is assumed as an independent (e.g. $x^a \in \mathcal{R}^{N\times 1}$; N observation from one variables) and the other as a dependent (e.g. x^b; N observation from another variable or a residual in section 4.3). The first step is to derive symbolic sequences as representatives of the statistical nature of the evolving dynamical system. For this purpose, partitioning of the time-series is required. Uniform partitioning (UP) and optimal partitioning (OP) can be selected for this purpose [94]. Also, maximum-entropy partitioning (MEP) [95] is based on maximizing the *Shannon* entropy of the symbolic sequence, thus the time-series regions with rich information have narrower partitions and those with sparse information have broader counterparts. All time-series in the simulation and industrial results of this chapter are partitioned utilizing (MEP) for constructing symbol sequences. As can be seen in Fig. 4.2, as a pictorial example of the partitioning and symbolic encoding of a time-series, the signal space Φ of a time-series is partitioned for instance into two number of mutually exclusive and exhaustive regions that are labeled as symbols $\sigma_i \in \Sigma,\ i = 0, 1$, i.e., the number of cells are identically equal to the cardinality of the alphabet (symbol) Σ. If the value of the time-series at a given instant is located in a particular cell, then it is coded with the symbol associated with that cell. Thus, a finite array of symbols s called a symbol string (see Fig. 4.2), can be generated from each (finite-length) time-series data (e.g. process variable x).

The next step is to generate the probabilistic finite state machines (PFSM) out of the symbolized time-series. To achieve this, four-string states $q_i \in Q,\ i = 0, ..., 3$ with length $D = 2$ are generated from all possible permutations of symbols. Hence, by considering a window with length $D = 2$ and moving it along the s, the time-series can be encoded.

To reformulate the derivations for a specific time-series x_a, the probability of being at state $q_i^{x_a}$ which is called state probability $p(q_i^{x_a}),\ q_i^{x_a} \in Q^{x_a}$ at the corresponding time epoch is one of the key components of the symbolic dynamic filtering. The state probability vector of a time-series is $P = [p(q_1^{x_a}), ..., p(q_{|Q^{x_a}|}^{x_a})]$, where $|Q^{x_a}|$ is the state cardinality for time-series x_a. As shown in Eq. (4.6), $\hat{p}(q_i^{x_a})$ can be estimated by frequency counting as the ratio of the number of times state $q_i^{x_a}$ occurs, i.e. $N(q_i^{x_a}) \rightarrow$ number of occurrence of $q_i^{x_a}$, in the symbol sequence over the number of times that all states $q_j^{x_a},\ j \in Q^{x_a}$ occur.

$$\hat{p}(q_i^{x_a}) = \frac{N(q_i^{x_a})}{\sum_{j\in Q^{x_a}} N(q_j^{x_a})} \tag{4.6}$$

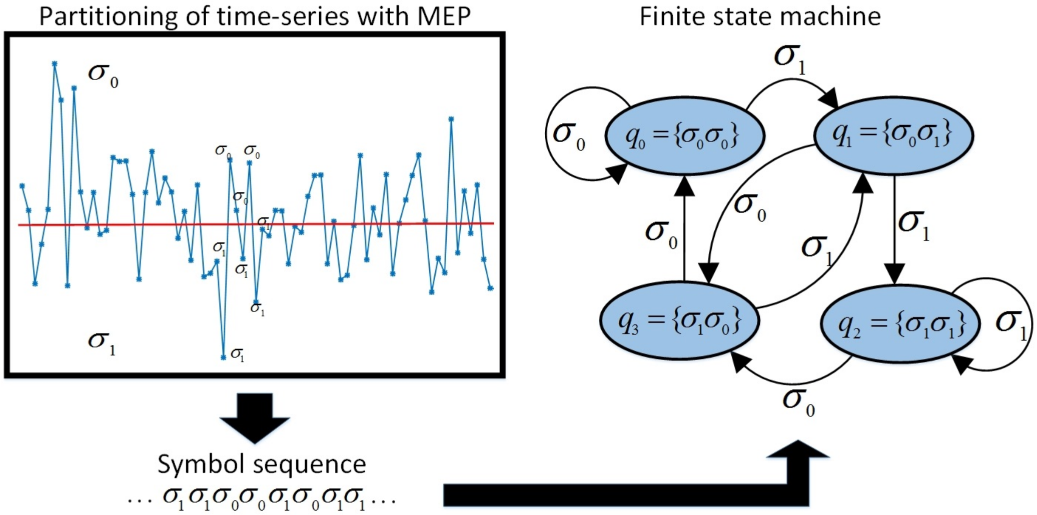

Partitioning of time-series with MEP
Finite state machine
σ_0
σ_1
σ_0 σ_0
σ_0
σ_1
σ_1
σ_1
σ_1
σ_1
$q_0 = \{\sigma_0\sigma_0\}$
$q_1 = \{\sigma_0\sigma_1\}$
$q_3 = \{\sigma_1\sigma_0\}$
$q_2 = \{\sigma_1\sigma_1\}$
σ_0
σ_1
σ_0
σ_0
σ_1
σ_1
σ_1
σ_0
Symbol sequence
$\ldots \sigma_1\sigma_1\sigma_0\sigma_0\sigma_1\sigma_0\sigma_1\sigma_1 \ldots$

Figure 4.2: Steps for construction of finite state machine from time-series.

Following the definition of state probability vector, the stationary irreducible state-transition probabilities $\pi_{ij}^{x_a x_a}$ is required at each epoch time to obtain the state transition matrix. $\Pi^{x_a x_a} = [\pi_{ij}^{x_a x_a} = p(q_i^{x_a}|q_j^{x_a})]$, $i, j = 0, ..., |Q^{x_a}|-1$, whose element represents transition probability from state $q_j^{x_a}$ to state $q_i^{x_a}$ upon occurrence of a symbol $\sigma^{x_a} \in \Sigma^{x_a}$ at each epoch time. In Fig. 4.3, a pictorial explanation of $\pi_{ij}^{x_a x_a} = p(q_i^{x_a}|q_j^{x_a})$ is shown. By utilizing the frequency counting method, the state transition probability can be determined as shown in Eq. (4.7).

$$\pi_{ij}^{x_a x_a} = p(q_i^{x_a}|q_j^{x_a}) = \frac{N(q_j^{x_a}, q_i^{x_a})}{\sum_{k=0}^{|Q^{x_a}|-1} N(q_j^{x_a}, q_k^{x_a})} \tag{4.7}$$

$N(q_j^{x_a}, q_i^{x_a})$ is the total counts of events when $q_i^{x_a}$ occurs adjacent to $q_j^{x_a}$ along the symbol sequence. For each time-series, a probabilistic finite state automata (PFSA) can be defined as following which describes the atomic dynamic characteristics of the time-series,

Definition 4.1 (PFSA [96]) *A probabilistic finite state automaton (PFSA) (describing time-series x_a) $\mathcal{A}_a^x$ is 4-tuple $\mathcal{A}^{x_a} = (\Sigma^{x_a}, Q^{x_a}, \Delta^{x_a}, \tilde{\Pi}^{x_a x_a})$, where :*

1. *$\Sigma^{x_a} = \{\sigma_0, \sigma_1, ..., \sigma_{|\Sigma^{x_a}|-1}\}$ is a nonempty finite alphabet (symbol) set with cardinality $|\Sigma|^{x_a} < \infty$.*

2. *$Q^{x_a} = \{q_0^{x_a}, q_1^{x_a}, ..., q_{|Q^{x_a}|-1}\}$ is a finite set of states with cardinality $|Q^{x_a}|$.*

3. *$\Delta^{x_a} : Q^{x_a} \times \Sigma^{x_a} \to Q^{x_a}$ is a state transition map.*

4. *$\Pi^{x_a x_a}$ is a square matrix of size $(|Q^{x_a x_a}| = |Q^{x_a}|^2)$; where $\pi_{ij}^{x_a x_a}$ is probability of moving from multi-dimensional joint state $q_i^{x_a x_a}$ at n^{th} epoch to $q_j^{x_a x_a}$ at $(n+1)^{th}$ epoch for $i, j = 0, ..., |Q^{x_a x_a}|-1$.*

5. *$\tilde{\Pi}^{x_a x_a} = [\tilde{\pi}_{ij}^{x_a x_a}] : Q^{x_a} \times \Sigma^{x_a} \rightarrow [0,1]$ is the $|Q^{x_a}|\times|\Sigma^{x_a}|$ symbol emission matrix (probability morph matrix), where $\tilde{\pi}_{ij}^{x_a x_a}$ is the probability of emitting a symbol $\sigma_j \in \Sigma^{x_a}$ from state $q_i \in Q^{x_a}$.*

The statistical properties of a PFSA can be described by the state transition matrix Π or the state probability vector P. For a symbol sequence s^{x_a}, it can be shown that the state probability vector $P = [p(q_0^{x_a}), ..., p(q_{|Q^{x_a}|-1}^{x_a})]$ is the left eigenvector of $\Pi^{x_a x_a}$ corresponding to the unique unity eigenvalue.

This research uses naive standard likelihood estimate method in Eqs. (4.6) and (4.7) which are based on frequency counting approach. In [97], it is shown that although we can replace the naive estimator with other counterparts, for the case of entropy estimation in the symbolic sequence, the naive approach has satisfactory results. The statistics of the symbolic sequence s^{x_a} represented by the state-transition $\mathbf{\Pi^{x_a x_a}}$ may change from one time epoch to another. Therefore, a suitable feature for this change is morph matrix $\tilde{\Pi}^{x_a x_a}$ indicating the symbol emission probability. The morph matrix elements $\tilde{\pi}_{ij}^{x_a x_a}$ is the probability of emitting a symbol $\sigma_j^{x_a} \in \Sigma^{x_a}$ from state $q_i^{x_a} \in Q^{x_a}$ [96], which is pictorially shown in the Fig. 4.3. Each entry of morph matrix $\mathbf{\tilde{\Pi}^{x_a x_a}}$ for a symbolic sequence can be determined by frequency counting as following,

$$\tilde{\pi}_{ij}^{x_a x_a} = \tilde{\pi}(q_i^{x_a}, \sigma_j^{x_a}) = \frac{N(q_i^{x_a}, \sigma_j^{x_a})}{N(q_i^{x_a})} \tag{4.8}$$

where $N(q_i^{x_a}, \sigma_j^{x_a})$ is the total count of events $q_i^{x_a} \in |Q^x|$ followed by $\sigma_j^{x_a} \in \Sigma^{x_a}$.

It should be noted that the assumption behind the SDF approach is that for the measured observations, symbolization is approximated as a *Markov* chain of order D (a positive integer), which is called *D-Markov* machine and required steps for determining D as a tuning parameter is explained in section 4.4.4.

In order to capture the cross dependency between two symbolic sequences s^{x_a} and s^{x_b}, which is an essential step in the estimation of conditional entropy $H({x_b}^{i+h\tau}|{x_b}^{l_1}, {x_a}^{l_2})$ in a window with the length of N symbols, relational PFSA defined as a *xD-Markov* machine is adopted to extract relational pattern(s) between time series [98].

Definition 4.2 (xD-Markov machine [98] [96]) *Let $\mathcal{A}^{\mathbf{x_a}}$ and $\mathcal{A}^{\mathbf{x_b}}$ be the PFSAs for the hypothetical source and target variables symbol streams s^{x_a} and s^{x_b}, Then a xD-Markov machine is defined as a 8-tuple $\mathcal{A}_{x_a \rightarrow x_b} \triangleq \{\Sigma^{x_a}, Q^{x_a}, \Sigma^{x_b}, Q^{x_b}, \Delta^{x_a}\ , \Delta^{x_b}, \Pi^{x_a x_b}, \tilde{\Pi}^{x_a x_b}\}$*

1. *$\Sigma^{x_a}, \Sigma^{x_b}$ are non-empty finite sets of alphabets belong to symbolic sequences s^{x_a}, s^{x_b}, respectively.*

2. *Q^{x_a}, Q^{x_b} are finite sets of states of the corresponding symbol sequences.*

3. *$\Delta^{(\cdot)} : Q^{(\cdot)} \times \Sigma^{(\cdot)} \rightarrow Q^{(\cdot)}$ is the general form of a state transition map which applies to every symbolic sequence involved in the calculation.*

4. $\tilde{\Pi}^{x_a x_b}$ *is the output symbol emission matrix of size* $(|Q^{x_a}|\times|\Sigma^{x_b}|)$*; where* $\tilde{\pi}_{ij}^{(x_a x_b)}$ *is probability of observing* $\sigma_j^{x_b} \in \Sigma^{x_b}$ *as the* $(n+1)^{th}$ *symbol in the sequence* s^{x_b}*, while making a transition from the multi-dimensional joint state sequence* q^{x_a} *at epoch* n^{th}*.*

Each entry of the relational morph matrix $\tilde{\Pi}^{x_a x_b}$ can be determined by following Eq. (4.8), where $\sigma_j^{x_b} \in \Sigma^{x_b}$.

4.4.3 Proposed Symbolic Dynamic Normalized Transfer Entropy (SDNTE)

The core idea behind the proposed root-cause fault diagnosis framework in this chapter is to find the process variable(s) that has the maximum contribution, i.e. maximum causal inference, to the deviation of the residual ψ calculated in section 4.3 from its normal zone (below its upper control limit), amongst all variables. The two time-series $x \in \mathcal{R}^N$ (e.g. one of the process variables time-series in a process under study) and $\psi \in \mathcal{R}^N$ (e.g. the residual signal determined using KPCA) are assumed as the driver and the driven, respectively. Then, as shown in Fig. 4.2, a symbol sequence is generated from each of the time-series data and their corresponding states are denoted by q^x and q^ψ, respectively. After partitioning and generating symbolic sequence and states, atomic (Def 4.1) and relational PFSAs (Def 4.2) for time-series x and ψ are determined.

According to Eq. (4.4) for calculation of $TE_{x \longrightarrow \psi}$, estimation of two conditional entropies $H(\psi_{i+h}|\psi_i^{l_1})$ and $H(\psi_{i+h}|\psi_i^{l_1}, x_i^{l_2})$ are required. For estimation of $H(\psi_{i+h}|\psi_i^{l_1})$, relational PFSA (*xD-Markov* machine, Def (4.2)) can be used, which represents the auto dependency of a time-series on its past values. However, the *xD-Markov* machine is not sufficient for measuring the dependency of a time-series (e.g. ψ) on past values of itself and past values of another time series (e.g. x), which is the case for $H(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2})$. Therefore, in order to estimate $H(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2})$, the new concept of the *joint xD-Markov* machine is proposed as follows,

Definition 4.3 (joint xD-Markov machine) *Let* $\mathcal{A}^x$ *and* $\mathcal{A}^\psi$ *be the PFSA with corresponding symbol streams* s^x *and* s^ψ*, respectively. Then a joint xD-Markov machine is defined as a 9-tuple* $\mathcal{A}^{x\psi\rightarrow\psi} \triangleq \{Q^x, \Sigma^x, \Delta^x, Q^\psi, \Sigma^\psi, \Delta^\psi, \tilde{\Pi}^{\psi\psi}, \Pi^{(x\psi)(x\psi)}, \tilde{\Pi}^{(x\psi)\psi}\}$

1. Σ^x*,* Σ^ψ*,* Q^x*,* Q^ψ *are determined as in (1)-(2) in Def 4.2.*

2. $\Delta^\psi : Q^\psi \times \Sigma^\psi \rightarrow Q^\psi$ *is a state transition map.*

3. $q_{r_1}^{\{x\psi\}} \in Q^{\{x\psi\}}$ *represents a joint state sequence similar to Def. 4.2 with* $r_1 = 0, ..., |Q^x|\times|Q^\psi|-1$*.*

4. $\tilde{\Pi}^{(\psi)\psi}$ *is the output symbol emission matrix of size* $|Q^\psi|\times|\Sigma^\psi|$*; where* $\tilde{\pi}_{ij}^{\psi\psi}$ *is probability of observing* $\sigma_j \in \Sigma^\psi$ *as the* $(n+1)_{th}$ *symbol in the sequence* s^ψ*, while making a transition from state* q_i *in symbol stream* s^ψ*.*

5. $\Pi^{(x\psi)(x\psi)}$ *is a square matrix of size* $\left(|Q^{\{x\psi\}}|= |Q^x|\times|Q^\psi|\right)$*, where* $\pi_{r_1 r_2}^{(x\psi)(x\psi)}$ *is the probability of moving from multi-dimensional joint state* $q_{r_1}^{\{x\psi\}}$ *at epoch* n^{th} *to* $q_{r_2}^{\{x\psi\}}$ *at* $(n+1)^{th}$ *for* $r_1, r_2 = 0, ..., |Q^{\{x\psi\}}|-1$*.*

6. *$\tilde{\Pi}^{(x\psi)\psi}$ is the output symbol emission matrix of size $(|Q^x|\times|Q^\psi|)\times|\Sigma^\psi|$; where $\tilde{\pi}^{(x\psi)\psi}_{ijk}$ is probability of observing $\sigma_k \in \Sigma^\psi$ as the $(n+1)_{th}$ symbol in the sequence s^ψ, while making a transition from state $\{q_i, q_j\}, q_i \in Q^x, q_j \in Q^\psi$ in joint symbol stream $\{s^x, s^\psi\}$.*

The pictorial definition of the join probability matrix $\Pi^{(x\psi)(x\psi)}$ and joint morph matrix $\tilde{\Pi}^{(x\psi)\psi}$ is shown in Fig. 4.3 for two general time-series $x_a := x$ and $x_b := \psi$. A joint state sequence needs to be generated and utilized for simultaneous measurement of cross-dependency between a time-series x and residual time-series ψ as well as its own auto-dependency on its past values. By using this definition, the conditional entropies required for the estimation of TE in Eq. (4.4) are defined as follows,

$SDH(\psi_{i+h\tau}|\psi_i^{l_1})$:

This conditional entropy refers to the ability to predict future $\psi_{i+h\tau}$ by using the past values of itself ${\psi_i}^{l_1}$. This ability is quantified by symbolic emission matrix $\tilde{\Pi}^{\psi\psi}$. This matrix quantifies the prediction capability (or uncertainty) of the next, i.e, $(n+1)_{th}$ symbol of the sequence s^ψ, from the known history of sequence till n_{th} time stamp (See Fig. 4.3). Therefore, by pursuing the conventional definition of *Shannon* entropy, the proposed symbolic dynamic-based entropy (SDH) is defined as follows,

$$\begin{aligned} H(\psi_{i+h\tau}|\psi_i^{l_1}) &= -\sum p(\psi_{i+h\tau}, \psi_i^{l_1}) \log p(\psi_{i+h\tau}|\psi_i^{l_1}), \\ \Rightarrow SDH(\psi_{i+h\tau}|\psi_i^{l_1}) &= -\sum_{k=1}^{|\Sigma^\psi|}\sum_{j=1}^{|Q^\psi|} p(q_j^\psi)\tilde{\pi}^{\psi\psi}_{jk} \log \tilde{\pi}^{\psi\psi}_{jk} \end{aligned} \tag{4.9}$$

where, $\tilde{\pi}^{\psi\psi}_{jk}$ is the morph matrix defined in Def (4.1) and can be determined using Eq. (4.8). $p(q_j^\psi)$ is state probability that can be estimated using Eq. (4.6).

$SDH(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2})$:

This joint conditional entropy refers to the ability of predicting future $\psi_{i+h\tau}$ by using both past values of ${\psi_i}^{l_1}$ and ${x_i}^{l_2}$. Intuitively, this ability is provided by proposed joint *xD-Markov* machine $\mathcal{A}^{x\psi\rightarrow\psi}$ and it is quantified by symbolic emission matrix $\tilde{\Pi}^{(x\psi)\psi}$, where each element is the probability of observing a particular symbol σ^ψ_{n+1} (at the $(n+1)_{th}$ time epoch) in the symbol stream s^ψ, given the joint process $\{x, \psi\}$ is in the state $q_n^{x\psi}$ at n^{th} epoch time. The state of joint process (symbol sequence) can be derived from the combination of two individual processes x and ψ as, $q^{x\psi} = \{q^x, q^\psi\}$. The symbolic dynamic-based version of this joint conditional entropy with regards to PFSA $\mathcal{A}^{x\psi\rightarrow\psi}$ is derived as following,

$$\begin{aligned} H(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2}) &= -\sum p(\psi_{i+h\tau}, \psi_i^{l_1}, x_i^{l_2}) \log p(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2}) \\ \Rightarrow SDH(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2}) &= -\sum_{k=1}^{|\Sigma^\psi|}\sum_{j=1}^{|Q^\psi|}\sum_{i=1}^{|Q^x|} P(q_i^x, q_j^\psi)\tilde{\pi}^{(x\psi)\psi}_{ijk} \log \tilde{\pi}^{(x\psi)\psi}_{ijk} \end{aligned} \tag{4.10}$$

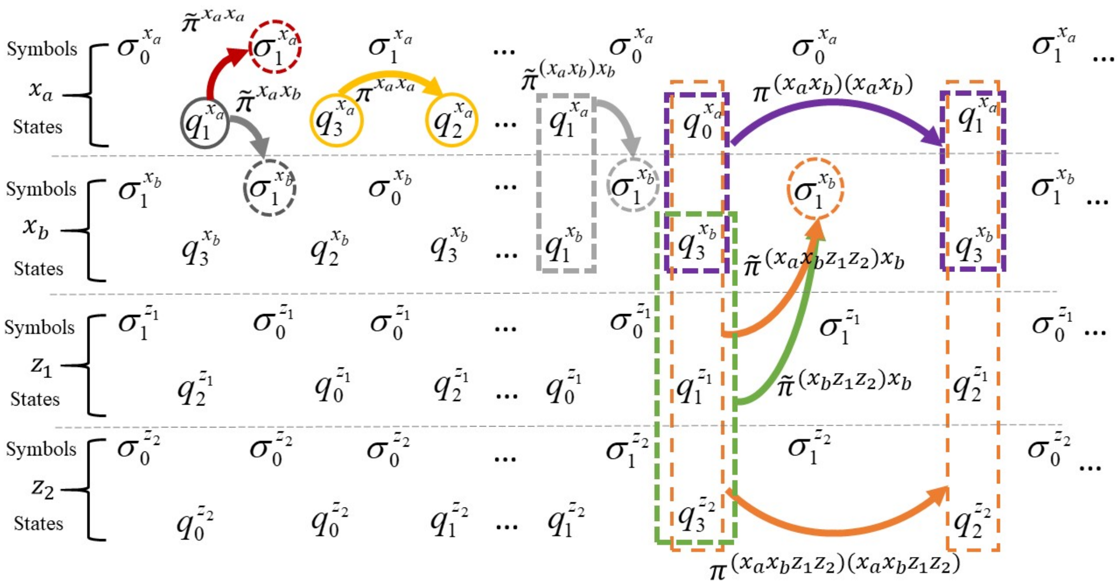

Symbols
x_a
States
Symbols
x_b
States
Symbols
z_1
States
Symbols
z_2
States
$\sigma_0^{x_a}$ $\sigma_1^{x_a}$ $\sigma_1^{x_a}$ ... $\sigma_0^{x_a}$ $\sigma_0^{x_a}$ $\sigma_1^{x_a}$ ...
$\tilde{\pi}^{x_a x_a}$
$\tilde{\pi}^{x_a x_b}$
$\pi^{x_a x_a}$
$\tilde{\pi}^{(x_a x_b) x_b}$
$\pi^{(x_a x_b)(x_a x_b)}$
$q_1^{x_a}$ $q_3^{x_a}$ $q_2^{x_a}$ ... $q_1^{x_a}$ $q_0^{x_a}$ $q_1^{x_a}$
$\sigma_1^{x_b}$ $\sigma_1^{x_b}$ $\sigma_0^{x_b}$ ... $\sigma_1^{x_b}$ $\sigma_1^{x_b}$ $\sigma_1^{x_b}$ ...
$q_3^{x_b}$ $q_2^{x_b}$ $q_3^{x_b}$ ... $q_1^{x_b}$ $q_3^{x_b}$ $q_3^{x_b}$
$\tilde{\pi}^{(x_a x_b z_1 z_2) x_b}$
$\sigma_1^{z_1}$ $\sigma_0^{z_1}$ $\sigma_0^{z_1}$ ... $\sigma_0^{z_1}$ $\sigma_1^{z_1}$ $\sigma_0^{z_1}$ ...
$q_2^{z_1}$ $q_0^{z_1}$ $q_2^{z_1}$ ... $q_0^{z_1}$ $q_1^{z_1}$ $q_2^{z_1}$
$\tilde{\pi}^{(x_b z_1 z_2) x_b}$
$\sigma_0^{z_2}$ $\sigma_0^{z_2}$ $\sigma_0^{z_2}$ ... $\sigma_1^{z_2}$ $\sigma_1^{z_2}$ $\sigma_0^{z_2}$ ...
$q_0^{z_2}$ $q_0^{z_2}$ $q_1^{z_2}$ ... $q_1^{z_2}$ $q_3^{z_2}$ $q_2^{z_2}$
$\pi^{(x_a x_b z_1 z_2)(x_a x_b z_1 z_2)}$

Figure 4.3: Pictorial explanation of the components explained in Def. 4.35.1. The colored broken rectangles indicate the joint states. All the proposed probabilities in section 4.4.3 are shown between the n^{th} epoch and $(n+1)^{th}$ epoch. However, for the sake of clarity, some of them are defined on other epochs.

where, $P(q_i^x, q_j^\psi)$ is the probability of joint symbolic sequence x be in the state $q_i^x \in Q^x$ and ψ be in symbolic state $q_j^\psi \in Q^\psi$, simultaneously at a given time instant. State probability distribution of the joint process $\{x, \psi\}$ is derived from combining states of two individual D-Markov machines, $\mathcal{A}^x$ and $\mathcal{A}^\psi$. The states of combined machine are represented by $q_{ij}^{x\psi} = \{q_i^x, q_j^\psi\}$. State transition matrix, $\Pi^{(x\psi)(x\psi)} = [\pi_{lm}^{(x\psi)(x\psi)}]$ is further extracted from the joint state sequence $Q^{x\psi}$, and it is of size $(|Q^x| \times |Q^\psi|) \times (|Q^x| \times |Q^\psi|)$. This proposed state transition matrix can also quantify the uncertainty in the joint process $\{x, \psi\}$.

Each element of the above mentioned state transition matrix is given by,

$$\begin{gathered}\pi_{ij}^{(x\psi)(x\psi)} = P(q^{(x\psi)}(n+1) = q_j^{(x\psi)} | q^{(x\psi)}(n) = q_i),\ \forall n, \\ \textstyle\sum_j \pi_{ij}^{(x\psi)(x\psi)} = 1\end{gathered} \tag{4.11}$$

where, $\{q_i, q_j\} \in Q^{x\psi}$. The state transition probability is calculated from the state sequence, by counting the number of transitions between each pair of states. The state transition matrix $\Pi^{(x\psi)(x\psi)}$ is the corresponding irreducible stochastic matrix of joint symbolic time-series, where each row sum is 1. The left eigenvector $p_{ij}^{x\psi}$ corresponding to the unique unit eigenvalue of the state transition matrix is the probability vector whose elements are the stationary probabilities of the states belonging to $Q^{x\psi}$.

By utilizing the proposed $SDH(\psi_{i+h\tau}|\psi_i^{l_1})$ and $SDH(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2})$, in order to determine the strength of the causal relation between two time-series, the normalized transfer entropy is defined as follows,

Definition 4.4 (symbolic dynamic-based normalized TE (SDNTE)) *The normalized TE from variable x to ψ by incorporating the proposed SDHs is defined as following,*

$$SDNTE_{x\longrightarrow\psi} = \frac{|SDH(\psi_{i+h\tau}|\psi_i^{l_1}) - SDH(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2}) - SDTE_{x\rightarrow\psi}^{training}|}{SDH(\psi_{i+h\tau}|\psi_i^{l_1})} \in [0,1] \quad (4.12)$$

where, $SDTE_{x\longrightarrow\psi}^{training}$ is defined as an average of $SDTE_{x\longrightarrow\psi} = SDH(\psi_{i+h\tau}|\psi_i^{l_1}) - SDH(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2})$ obtained on n_s trials of time-series in training data-set (i.e. The same off-line training data-set in normal operating condition, which was used for generating the base-line of the residual ψ). Eq. (4.12) intuitively represents the fraction of information about the future observation of ψ obtained from both past values of x and ψ, after discarding, first, the information about the future of ψ provided by its own past, and second, the calculation bias for normal operating condition which is represented by $SDTE_{x\rightarrow\psi}^{training}$. In the definition of Eq. (4.12), the zero entropy is considered as the origin and represent a deterministic ψ. This proposed SDNTE will be used later in section 4.5 for root-cause fault diagnosis purposes.

4.4.4 Parameters Determination

- **Cardinality of Symbols ($|\Sigma|$):**

Consider $H(k-1) = -\sum_{i=0}^{i=k-1} \mathcal{P}_i log \mathcal{P}_i$ as the *Shannon* entropy of the symbolic sequence obtained from partitioned data with k symbols, where, $\mathcal{P}_i$ is the i^{th} symbol probability in a symbolic sequence with length N^{SD} (number of samples required for the proposed SDNTE). The following steps are proposed to obtain the optimal symbol cardinality for a time-series [99],

Steps:
1) Set $k = 2$ and choose a threshold ϵ_h such that $0 < \epsilon_h << 1$,
2) Sort the data with length N^{SD} samples in the ascending order,
3) Partition the raw data into a symbolic sequence as shown in Fig. 4.2,
4) Determine the symbols' probabilities $\mathcal{P}_i = \dfrac{N(\sigma_i)}{\sum_{j=0,...,k-1} N(\sigma_j)}$, $i = 0, 1, ..., k-1$,
5) Compute $H(k-1) = -\sum_{i=0}^{i=k-1} \mathcal{P}_i log \mathcal{P}_i$ and $h(k-1) = H(k-1) - H(k-2)$,
6) If $h(k) < \epsilon_h$, then stop and set $|\Sigma|= k$, else increment k by 1 and go to step 3.

In the above algorithm, ϵ_h is a design parameter that must be selected according to the level of noise in time-series. It should be noted that a very small value of ϵ_h leads to a large number of symbols, thus, increases the number of states. On the other hand, a large choice of ϵ_h may cause insufficient partitions in *D-Markov* machines with regards to the intrinsic dynamic of time-series. In the training step, the cardinality of symbols $|\Sigma|$ are determined and will be used for the real-time root-cause fault diagnosis procedure.

- **Depth of Markov Machine (D):**

This parameter has a crucial role in generating the *D-Markov* machines since it exponentially increases the maximum number of states (i.e. $|Q|\leq |\Sigma|^{D}$). In general, a very small choice of D leads to insufficient memory of the *D-Markov* machine, thus, lose of information about the dynamic of the time-series. On the other hand, large value of D increases the sensitivity to dynamic distortion and computational complexity. One way to determine the proper value for the depth D is monitoring the entropy rate while changing D [99]. The rate of entropy h_μ is shown in Eq. (4.13) for a symbolic stochastic process of time-series x, which also can be interpreted as the uncertainty in the next symbol.

$$h_\mu = \sum_{i=1}^{|Q^x|}\sum_{j=1}^{|\Sigma^x|} p(q_i^x)\tilde{\pi}_{ij}^{xx} log \tilde{\pi}_{ij}^{xx} \tag{4.13}$$

As a rule of thumb, h_μ monotonically decreases while the depth D increases. However, beyond a certain point, increasing D does not significantly change the entropy rate, hence, the corresponding D is the optimal choice for the depth of *D-Markov* machine. It should be noted that for noise-free time-series $h_\mu \longrightarrow 0$ and for noisy time-series, h_μ monotonically decreases to a small non-zero value.

•The State Cardinality ($|Q|$):

The depth D of an *xD-Markov* machine exponentially increases the number of states $|Q|= |\Sigma|^D$. Therefore, for processes with relatively high depth and symbols' cardinalities, it is required to truncate those states with a relatively small probability of occurrence. In [99], it is proposed to define a threshold $\varepsilon = 1/N^{SD}$ based on the length of the symbols N^{SD} such that if the probability of some states is less than ε, they are considered as a transient state and can be neglected. Also, the number of states can be further reduced by the state-merging algorithm [99] which merges the state with the same probability of happening. State splitting is another method [100] for adjusting the number of PFSAs states by using a metric on the probability distributions of symbolic blocks.

• Required Number of Samples N^{SD}:

One of the advantages of proposed SDTE is that the number of samples N^{SD} (i.e. length of symbolic sequence in Fig. 4.2) required for generating the state transition matrix is less than the number of samples N^{kernel} required for estimating the entropies using the conventional KDE utilized in [59] [57]. In [101] [99], two methods based on the properties of state transition matrix and inference approximation with specified absolute error ϵ and probability λ are proposed. In the simulation result section 4.5 of this chapter, the former method that is based on Frobenius theorem is utilized for determining N^{SD} that is the minimum required length of the symbolic sequence s for time-series x and ψ. Later the number of temporal samples N^{SD} required for SDNTE estimation is compared with its counterparts N^{kernel} and it is shown that the proposed SDNTE method requires less amount of temporal data (i.e. $N^{SD} < N^{kernel}$) in a similar condition.

4.4.5 Computational Complexity

Estimation of TE using joint PDFs commonly involves heavy computation for causality analysis. In [59], joint PDFs are estimated using multi-dimensional kernel functions. The computational order of a q dimensional PDF is $O(N^2q^2)$, where $q = l_1 + l_2 + 1$ is determined by the number of sample delay incorporated into embedding vectors in Eq. (4.4). Thus, the total complexity order of calculating TE using Eq. (4.4) is $O(N^2(l_1 + l_2)^2)$.

One of the contributions of the proposed SDNTE approach is the improvement in the computational complexity, hence, real-time application of the TE for root-cause diagnosis. The complexity order of SDNTE estimation is divided into two parts: first, finding the computational complexity of the proposed conditional SDHs in Eqs. (4.10) and (4.9), and second, determining the computational complexity of the proposed symbolic dynamic normalized TE in Eq. (4.12).

In this chapter all the frequency counting formula that are proposed to estimated the state probability and morph emission matrices are calculated using a nested loop search approach. Therefore, the complexity order of probability state vector $P(q_i^x, q_j^\psi)$ inside the cascade summation in Eq. (4.10) is $O(N|Q^x||Q^\psi|)$. Also, the computational order for determination of $\tilde{\Pi}^{(x\psi)\psi} = [\tilde{\pi}_{ijk}^{(x\psi)\psi}]$ is of $O(N|\Sigma^\psi|)$. Thus, the total computational order for $SDH(\psi_{i+h\tau}|\psi_i^{l_1}, x_i^{l_2})$ is $O(N|\Sigma^x|^D \; |\Sigma^\psi|^{D+1})$. Similarly, the computational order for the Eq. (4.9) is of $O(N|\Sigma^\psi|^{D+1})$ and since it is less than the computational order of the proposed symbolic conditional joint entropy, the total computational order for the SDNTE proposed in the Eq. (4.12) is $O(N|\Sigma^x|^D \; |\Sigma^\psi|^{D+1})$. By simple comparison, even for the same amount of temporal data (i.e. $N^{kernel} = N^{SD}$) the proposed SDNTE needs less computational effort rather than its opponent kernel estimation of PDFs adopted in [57] and [59]. This difference is further presented in the simulation result in the section.

4.5 Simulation Results

In this section, a case study is conducted on the Tennessee Eastman process (TEP) which is a well-known benchmark for fault detection/diagnosis analysis. TEP contains 12 manipulated variables and a total of 41 measurements which include 22 intermediate variables and 19 quality indices. However, as shown in *Appendix*, Table 1, only 22 direct process measurements as well as 11 manipulated variables are considered as the process variables under study [50]. The flow chart of the TE process is also brought in *Appendix*, Fig. 2 for better understanding of the interconnections of the process variables.

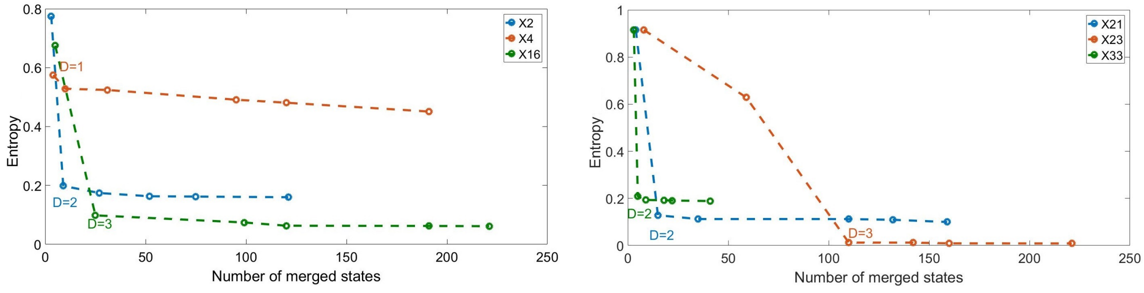
In TEP, 21 different malfunction scenarios are defined [102], which (IDV(1-15)) are the known faults and brought in *Appendix*, Table 2. Amongst these known faults, IDV(8), IDV(10) and IDV (11) which satisfy quasi-stationary criteria are considered for evaluation of the proposed framework.

4.5.1 Off-Line Training and Determination of Parameters for Tennessee Eastman:

The TEP is run for 72 hours and 72000 samples are collected for each scenario. $N = 500$ samples are collected from normal operating condition IDV(0) and the kernel function $k(x_i, x_j) = exp\left(-\frac{||x_i - x_j||^2}{3}\right)$ is utilized for conducting kernel trick, hence, generating the mean-centered kernel matrix K in Eq. (4.1). Then, after finding the covariance matrix C^Y, the cumulative percentage of variance (CPV=92%) is considered to extract 17 non-zeros singular values $\hat{\Lambda}$ and their corresponding principal directions $\hat{U}$. The confidence level of 98% is considered for determining the UCL_ψ.

The next step is to find the parameters of the proposed SDNTE method by using an initial $N^{SD} = 1500$ samples of data in normal conditions. Later, the minimum number of samples required for the real-time calculation will be determined. MEP method [95] is utilized to partition the TEP time-series for IDV(0). In Table 3, the calculated symbols' cardinality $|\Sigma|$, depths D and the number of states after merging are presented for all process variables and the residual signal. The number of symbols $|\Sigma|$ are found by deeming a threshold ϵ_h and following the steps presented in section 4.4.4. After finding the symbols' cardinality, the number of states $|Q|$ after merging and truncating the ignorable transient ones are determined for each process variable. Also, by monitoring the entropy rate $h_\mu(k)$ for all process variables, the depth D for constructing the *Markovian* machines is determined.

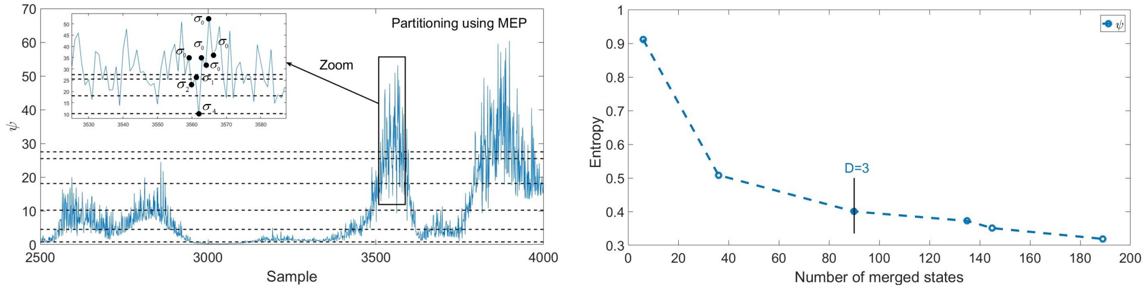
Figs. 4.4(a) and 4.4(b) indicate the monotonically decreasing trend of entropy rate $h_\mu(k)$ with respect to the D and number of merged states for five time-series X2-4-16-21-23-33. The optimum depths D are considered for variables when the $h_\mu(k)$ values remain relatively constant.



(a)

(b)

Figure 4.4: Entropy rate trend of the process variables for different values of depth D.



(a) The temporal data chosen for finding the parameters of the ψ and the pictorial procedure for partitioning and encodeing the ψ time-seires

(b) Trend of entropy rate for different values of depth D

Figure 4.5: Procedure details to find parameters of the D-Markov machines for time-series ψ.

Due to the fact that the residual signal supposedly behaves as random i.i.d noise for the ideal normal operation, whereas has a distinct characteristic (e.g. range of variation) for the faulty condition, the same normal set of data IDV(0) may not be an appropriate choice for determining the optimum parameters for PFSAs construction of the residual ψ. To this aim, $N^{SD} = 1500$ samples of the residual signal obtained for IDV(13) (i.e. a random choice among available quasi-stationary scenarios) is used for finding ψ parameters. Fig. 4.5(a) shows the temporal data chosen for parameter identification of ψ. Also, Fig. 4.5(b) indicates the trend of entropy rate for the residual signal with respect to the depth and corresponding merged states, where the optimum depth $D = 3$ is chosen.

As mentioned before, the initial number of symbol sequence $N^{SD} = 1500$ is considered for determining the parameters in Table 3. Then, by following the procedure in [99], the maximum number of required samples for X23 which needs more samples in comparison with other variable is calculated $N^{SD} = 900$. As a result, for the real-time root cause fault diagnosis, 900 samples of temporal data for each time-series are utilized for conducting the root-cause fault diagnosis.

4.5.2 Real-Time Root-Cause Fault Diagnosis in Tennessee Eastman Process:

After determining the optimum parameters of the proposed *D-Markov* machines, the proposed SDNTE is used to find the root-cause(s) for three quasi-stationary fault scenarios IDV(8), IDV(10) and IDV(11).

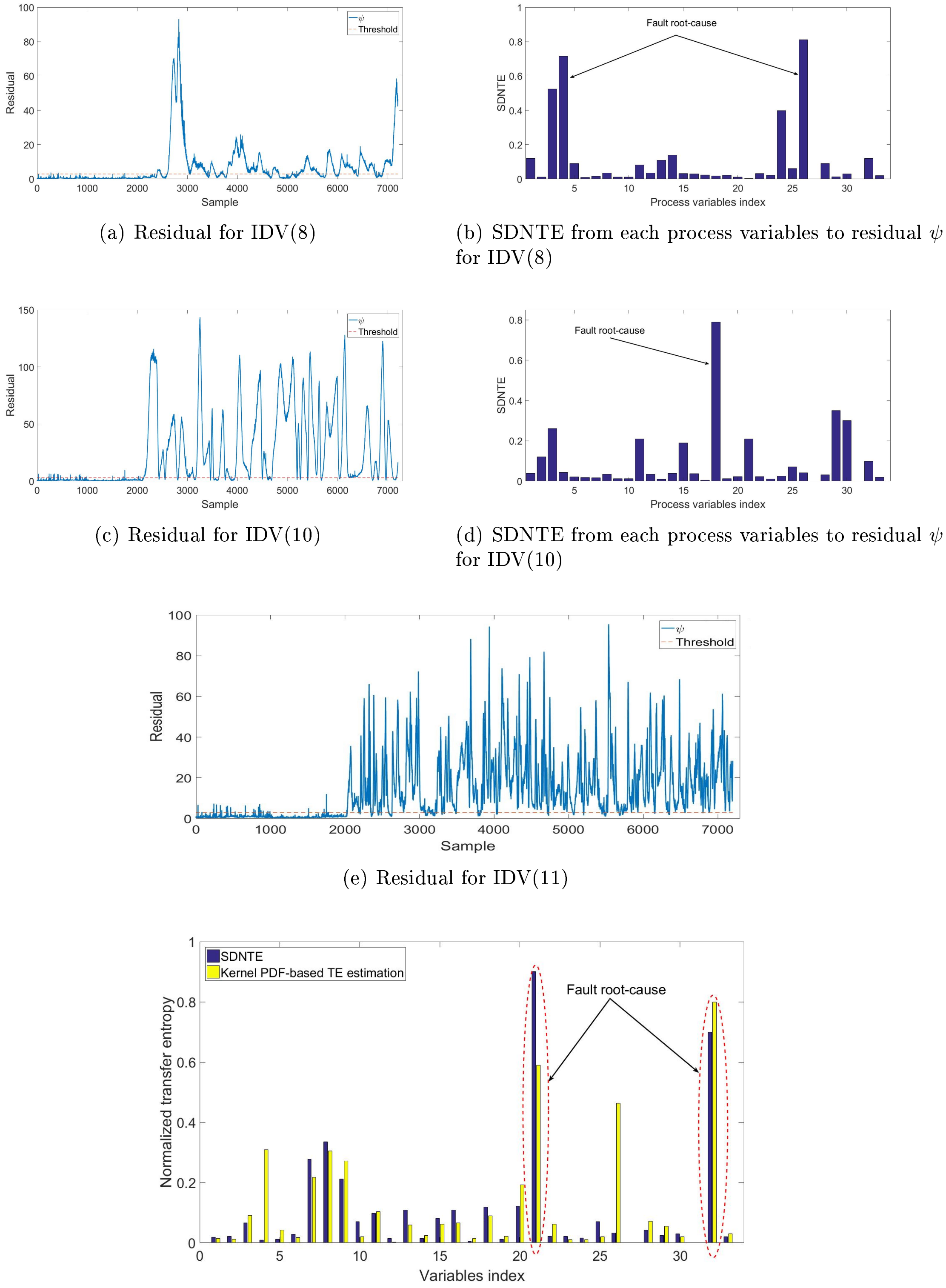
Fig. 4.6(a) shows the residual signal ψ for IDV(8) which is a random variation in A, B and C streams starting at 2000^{th} sample. After detecting fault at 2250^{th} sample, a window of temporal data between samples 2250 and 3150 are considered for time-series of all 33 process variables and residual signals. Then, the symbolic dynamic normalized transfer entropy (SDNTE) proposed in Eq. (4.12) is calculated from each process variables to the residual signal. As can be seen in Fig. 4.6(b), X4 (total feed-stream 4) and X26 (E feed-stream 4) have the top two contributions in residual signal. Also, X3 (E feed-stream 3) and X24 (E feed flow-stream 3) have the third and fourth ranks in the bar charts while the rest of the contributions are relatively negligible. With regard to the TE process flowchart in Fig. 2, if a fault is introduced into A/B/C composition, the controller (X26) is responsible to compensate the fault and accordingly change the total feed (X4) to adjust the set-point, thus, X4 and X26 are root-causes for IDV(8). The reason for relatively high contributions of X3 and X24 is that they are responsible to adjust the E feed flow along the controller X26 with delay to maintain the overall process performance, thus, any change in A/B/C stream directly affects them.

Fig. 4.6(c) indicates residual ψ for another random fault scenario IDV(10), which introduces variation into C feed temperature and directly affects the gas mixture temperature in stream 4. In Fig. 4.6(d), the causality of each process variable in ψ is shown and X18 has the only dominant contribution for generating the residual and can be considered as the true root-cause of fault. With regards to the nature of the IDV(10), stripper temperature X18 is the first variable that must react to any change in the C feed mixture.

To further demonstrate the computational effectiveness of the proposed SDNTE algorithm over the ordinary kernel PDF-based method for transfer entropy estimation [57] [59], the conditional entropies defined in Eq. (4.12) is determined using both methods and the results are presented in Fig. 4.6(e). The fault IDV(11) is considered for this comparison study which is a random variation of the reactor cooling water inlet temperature. This variation directly propagates into the reactor cooling water flow X32 and reactor cooling water outlet temperature X21 because of the closed-loop regulation (see Fig. 2). Moreover, other reactor sensors (X5-X6-X7-X8-X9), as well as the stripper sensors (X15-X16-X18-X30), can be significantly affected by this random variation and make the root-cause diagnosis even more arduous. The result for the causality contribution of each process variable to ψ is shown in Fig. 4.6(f). The contributions of variables X32 and X21 are correctly determined maximum amongst others by using both SDNTE and ordinary kernel PDF-based TE, thus, highlighted as the fault root-cause for IDV(11).

For the ideal case, the contribution values found by SDNTE and ordinary kernel PDF-based transfer entropy estimation method shown in Fig. 4.6(f) should approximately match for each process variable. However, due to the generic difference in the calculation procedure of each method and difference in the length of utilized temporal data, the results are slightly different, while the major contributors are consistent which leads to the same root-cause fault diagnosis conclusion. In Fig. 4.6(f), the contributions are consistent from both methods for X4 (total flow of stream 4) and X26 (flow controller in stream 4) has a questionable dissimilarity. The reason behind this contrast may be due to the difference in the $N^{SD} = 900$ and $N^{kernel} = 2000$ such that for calculation of the yellow bars, wider temporal data are used which include the delayed feedback dependency between reactor temperature and stream 4.

In order to show the computational advantage of the proposed SDNTE method over the ordinary kernel PDF-based estimation algorithm for transfer entropy (see [57] [59]), the total



(a) Residual for IDV(8)

(b) SDNTE from each process variables to residual ψ for IDV(8)

(c) Residual for IDV(10)

(d) SDNTE from each process variables to residual ψ for IDV(10)

(e) Residual for IDV(11)

(f) SDNTE from each process variables to residual ψ for IDV(11)

Figure 4.6: Root-cause fault diagnosis result of TEP for the fault scenario IDV(8).

time delay is compared in the same calculation condition (CPU i7 @3.2 GHZ and RAM 8 GB) for fault IDV(11). The time delay for conducting kernel trick and detecting fault which is similar for both cases is 0.65 second. The time delay for root-cause fault diagnosis utilizing proposed fast SDNTE is 16.2 second and for ordinary kernel, the PDF-based method is 68.9 second, which indicates more than 4 times improvement in real-time calculation speed.

4.6 Industrial Application

In this subsection, the data from a large scale centrifuge is used to test the applicability and performance of the proposed root-cause fault diagnosis scheme in industrial processes. The centrifuge understudy is shown in Fig. 4.7(b) is fed with a mixture of liquid containing oil and water as well as other solid substances and its main task is to separate water and solids from oil by using centrifugal force. This centrifuge is assumed as a complex mechanical process that the inner structure and variables relations are unknown to the operators which hinder the application of model-based monitoring techniques. The measurements for this separation process are; vibration RMS, heavy phase pressure (KPa), Eline pressure (KPa), power (V) and lubrication oil pressure (KPa) and inputs are rotating speed (RPM) and production flow (Lit/sec), which are all considered as the process variables.

There are two fault scenarios that site engineers have encountered with the centrifuge. First, *nozzle plugging* that happens when there are residue sediments accumulating at one or some of the nozzles in the bowl of the centrifuge, which also create excessive centrifugal force acting as imbalance around the rotating shaft. This type of fault makes the vibration RMS reach its critical value leading to the shut-down of the process. Severe vibration caused by this process fault might lead to cracks in the centrifuge's shaft and consequently breakdown. The second type of fault detected by operators is power fluctuation that leads to a change in speed and creates symptoms similar to nozzle plugging and can not be diagnosed by common monitoring techniques. The main challenge in this process is that both regulatory change of speed and true presence of nozzle plugging have the same impact on residual ψ. However, the proposed scheme can detect and separate input changes from malfunction and further isolate the type of existing aforementioned faults utilizing the operator's knowledge shown in Fig. 4.7(a). For instance, for the case of *nozzle plugging*, the key monitoring variables are vibration RMS and E-line pressure. These two variables can be treated as key fault indicators (KFI). These KFIs are used along with the proposes scheme to isolate the true plugging events.

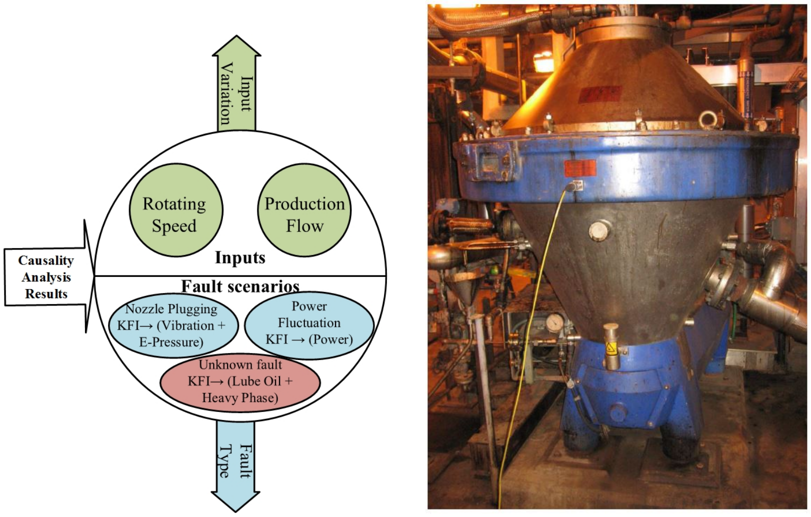
The industrial data was collected with a 1 second sampling interval that is a relatively high sampling frequency with respect to the time constant of the process, thus, the data is down-sampled with a ratio of 10:1 for conducting the proposed SDNTEs. N=2000 samples of down-sampled data are used for training the kernel PCA base-line and a standard deviation of 5 is considered for the Gram kernel matrix. To create residual ψ, 9 principal directions are determined using $CPV = 85\%$. For the determination of the PFSAs machines' parameters, the optimum number of samples $N^{SD} = 1000$ is considered and results are presented in Table

Table 4.1: The parameters for PFSA construction of the industrial centrifuge variables

Variables	ϵ_h	$\lvert\Sigma\rvert$	D
Lubrication Oil	0.2	4	1
Power	0.2	7	2
Speed	0.2	6	2
Vibration RMS	0.1	8	2
Production Flow	0.2	5	2
Heavy Phase	0.2	5	1
Eline	0.2	5	2
ψ	0.1	10	2

Table 4.2: Scores for each fault scenario determined by the summation of corresponding SD-NTEs with regards to the operator's knowledge

	S_{speed}	$S_{prodflow}$	$S_{F_{nozzle}}$	$S_{F_{power}}$	$S_{F_{unknown}}$
12 Feb	0.294	0.030	**0.621**	0.019	0.036
15 Feb	0.214	0.079	**0.527**	0.055	0.124
16 Feb	**0.539**	0.09	0.14	0.075	0.154
22 Feb	**0.551**	0.04	0.149	0.015	0.029
23 Feb	0.211	0.069	0.161	0.115	**0.244**
23 Feb	**0.511**	0.001	0.101	0.001	0.358
24 Feb	0.001	0.001	0.152	**0.84**	0.006


Input Variation
Rotating Speed
Production Flow
Inputs
Causality Analysis Results
Fault scenarios
Nozzle Plugging KFI→ (Vibration + E-Pressure)
Power Fluctuation KFI → (Power)
Unknown fault KFI→ (Lube Oil + Heavy Phase)
Fault Type

(a) Operators knowledge

(b) Centrifuge

Figure 4.7: Industrial centrifuge.

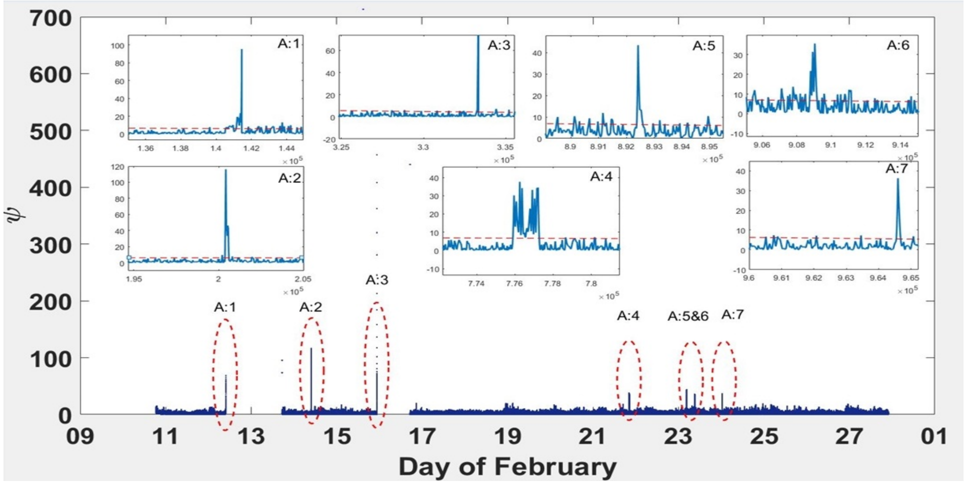


Figure 4.8: The residual ψ for centrifuge data in February. The exaggerated windows indicate the temporal data portion before down-sampling utilized for calculating the SDNTE from process variables to residual ψ.

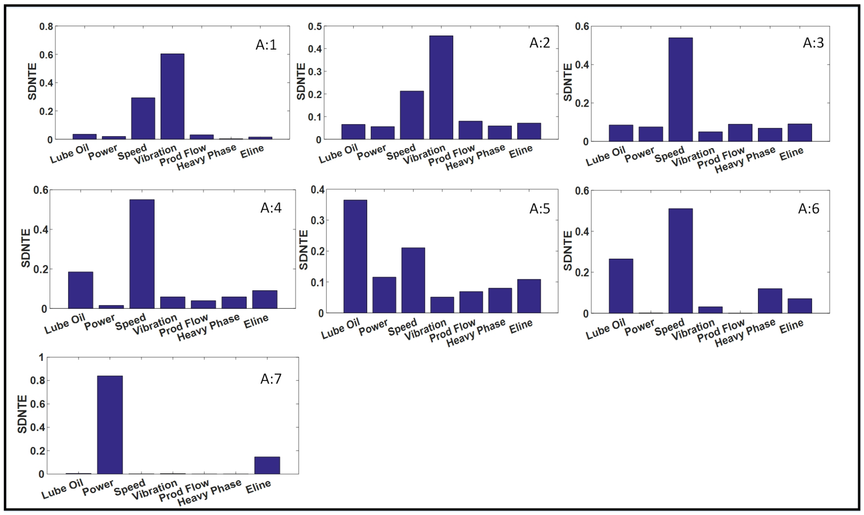


Figure 4.9: Contribution of all seven process variables for each detected fault event using proposed SDNTE method.

4.1.

Fig. 4.8 plots the down-sampled residual signal against days for the month of February, where, the fault events are highlighted through dotted elliptical shapes and the raw data in the vicinity of highlighted areas are magnified and shown in subplots of Fig. 4.8. With respect to the nature of the centrifuge data, for calculating the $SDNTE_{x \longrightarrow \psi}$ for each fault event, a window of length $N^{SD} = 1000$ is chosen from down-sampled time-series starting from 500 samples before the violation of upper control limit. The SDNTEs from seven process variables to residual signal for events A:1-7 are presented in Fig. 4.9 as bar charts. Missing intervals in the residual in Fig. 4.8 are due to the dates that the centrifuge was shut down maintenance purposes.

For this industrial application, the operators can utilize the SDNTE result for diagnosing the pre-defined fault scenarios shown in Fig. 4.7(a) by summation of the highest contributions. According to the operators' knowledge shown in Fig. 4.7(a), the $SDNTE_{x_i \longrightarrow \psi}$ of variables corresponding to each fault scenario are summed and presented in the Table 4.2 for decision making. Accordingly, for each event in Table 4.2, the maximum value is identified as the type of fault and was matched with the inspection results.

4.7 Summary

In This chapter, we propose a novel root-cause fault diagnosis framework which has efficient calculation complexity and requires less number of TE calculation for conducting causality analysis compared to existing techniques. The underlying idea behind the proposed framework is to measure the strength of the contribution of process variables towards the change in residual signal once the fault is detected. To find these causal dependencies of each process variable $x \in X \in \mathcal{R}^{m \times N}$ on the residual $\psi \in \mathcal{R}^N$, we propose a new and fast technique named symbolic dynamic normalized transfer entropy (SDNTE). Intuitively, proposed $SDNTE_{x \longrightarrow \psi}$ can be considered as a quantity gauges the level of contribution of each process variable to the existing fault in a process. The SDNTE enables the real-rime application of transfer entropy for root-cause identification which has been suffered from computational complexity. In the end, the proposed strategy is applied to the Tennessee Eastman benchmark and the results are compared with the conventional KDE method for estimating the transfer entropy. Also, the root-cause fault is identified by conducting the proposed strategy on an industrial centrifuge, which endorses its application for complex industrial process monitoring and diagnostics.

Chapter 5

Autonomous Root-Cause Fault Diagnosis

5.1 Introduction

Autonomous operation and adaptation is an active line of research that has attracted great interests in manufacturing and process industries. The word "Autonomous" can be used when a manual task is performed automatically by algorithms and the level of human interaction is either reduced or eliminated. Among various existing applications, automatic parameter tuning has raised significant attention in the field of control and robotics [103] [104] [105]. Autonomous algorithms are in place not only to remove manual parameter tuning but also enable the real-time implementation of an approach without dependency on operators' choice of actions. For example, in [106], an approach is proposed to dynamically update the clusters upon receiving new data on the data-streaming platforms. For this particular line of research, there are various parameter-dependent approaches (e.g. evolving clustering methods ESOMs [107], growing neural gas [108], batch clustering methods k-means [109], etc.) that can cluster the live streaming data. However, in the sense of automation, the proposed autonomous approach in [106] can be seen as an important improvement towards a *a priori* knowledge-free (or at least towards a parameter-sensitive-low) evolving clustering.

Autonomous algorithms can be applied in process monitoring applications such as fault diagnosis, which usually suffers from heavy tuning and require operators' knowledge. Root-cause fault diagnosis is considered as one important step for the process monitoring because it can reveal the source of the detected malfunction. Knowing the fault root-cause provides sufficient information for operators to adopt proper maintenance actions. In [58], TE and DTE are utilized to find the information pathway among process measurements based on operator/expert knowledge. Similarly, Ma *et al* [64] used a combination of TE and DTE for finding the pathway among process measurements, but all the intermediate variables are chosen according to the logics provided by process operator. By considering the power of autonomous algorithms and the ongoing challenges for root-cause fault diagnosis using TE/DTE, a framework for autonomous process root-cause fault diagnosis is proposed in this chapter.

In Chapter 4, symbolic dynamic filtering was utilized to develop a computationally efficient scheme for TE estimation of time-series. This scheme is further extended in this chapter to handle DTE estimation with intermediate variables.

5.2 The Proposed Framework

The following lists main challenging problems to be addressed in this chapter:

(1) High computational complexity of DTE with the presence of multiple intermediate variables.

(2) Determination of intermediate variable(s) required for DTE without human intervention. Since time-complexity of DTE rises exponentially with the number of intermediate variables, proper determination of intermediate variables plays a crucial role in the proposed strategy;

(3) Fully automated strategy after collection of time-series data to systematic identification of the root-cause variable(s) for the detected fault.

Therefore, the primary goal of the proposed work in this chapter is to autonomously locate root-cause fault variable(s) according to the causal information pathways amongst process measurements under the faulty condition without the need for knowledge of process topology and intervention of an expert. A schematic diagram of the overall framework is shown in Fig. 5.1, with each component and main steps briefly summarized in the following. Detailed design and analysis are presented in the subsequent sections.

Step (1): A fault detection method (e.g. KPCA for non-linear cases and PCA for linear cases) is utilized to generate a monitoring index. It is worthwhile to mention that the proposed strategy in this chapter can be used for both linear and non-linear processes.

Step (2): Upon detection of a fault, a fast "screening" and preliminary diagnosis procedure (e.g. RBC [110] or ACRC [2]) is used to identify and flag the potential faulty variables. In other words, at this stage, only a portion of the variables are selected as the candidates for further root-cause fault diagnosis. It is assumed that the process fault affects at least one measurement variable for the sake of detectability and diagnosability.

Step (3): Symbolic dynamic-based normalized transfer entropy (SDNTE) proposed by authors in [20] are computed for all pairs of potential faulty candidates from step (2) to generate an initial causal graph.

Step (4): Since certain connections in the initial causal graph may be spurious/indirect, a novel symbolic dynamic-based normalized direct transfer entropy (SDNDTE) approach with multiple intermediate variables (IVs) is proposed to eliminate the spurious and/or indirect connections. To address the challenge (1), the SDNDTE is proposed as a time-efficient alternative to calculating the direct transfer entropy (DTE). This new contribution enables the real-time application of DTE for root-cause fault diagnosis in process industries and other applications.

Step (5): In order to efficiently choose intermediate variable(s) required for calculation of

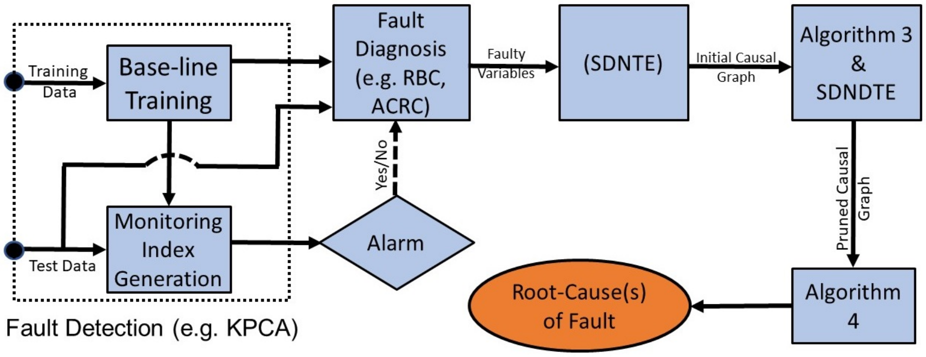


Figure 5.1: Schematic diagram of the proposed paradigm for autonomous root-cause fault diagnosis.

SDNDTE in step (4) which is also a solution to the challenge (2), in this step, IVs are classified into *Immediate IVs* and *Source IVs* (see Defs 5.3-5.2). Then two algorithms are proposed (see *Algorithms* 3 and 4) to efficiently determine both types of IVs by calculating SDNDTE for a pair of hypothetical source and target variables. Based on this, the initial causal graph is pruned by removing indirect/spurious causal path(s).

Step (6): To tackle challenge (3), *Algorithm* 5 is proposed for the pruned causal graph to autonomously locate the source (i.e. root-cause representative) of the detected fault.

5.3 Causal Structure and Identifiability of Causal Model

Each process can be represented by a set of structural equation models (SEMs) including the endogenous process measurements $X = [x_1, x_2, ..., x_m] \in \mathcal{R}^{N\times m}$ and exogenous disturbance variables $U = [u_1, u_2, ..., u_m] \in \mathcal{R}^{N\times m}$ that are independent from each other as the following,

$$x_i = g_i(f_i(PA(x_i)) + u_i), \;\; i = 1, ..., m \tag{5.1}$$

where g_i is invertible and f_i is either a linear or non-linear function. $PA(x_i)$ represents all the parent variables of x_i, also considered as the causes of x_i. As mentioned in [111], the concept of SEM can be also interpreted as *functional model classes.* In this research, the *identifiable functional model classes* (IFMOCs) are of interest. They are considered as underlying models for the process in order to show the identifiability of a unique causal graph $\bar{G}$.

Challenge: *Given a process data-set with sufficient i.i.d sample observations and sufficient number of measurements (V), let $P^{(x_i)}, i \in V$ represent the conditional probability distribution among process time-series. Assume that a direct acyclic graph (DAG) $\bar{G}$ exists and it represents the true causal structure of the temporal process. Is such a unique $\bar{G}$ identifiable by using set of $P^{(x_i)}s$?*

This question is important since identifiability is required as an essential assumption for the proposed strategy.

Identifiability of $\bar{G}$ for a given temporal process can be studied by checking the *Markov* and faithfulness conditions [112] [113] [114]. If these two conditions are satisfied, an algorithm such as PC (named after its authors, Peter and Clark) [115] can be utilized to partially reconstruct graph G which can be recovered up to Markov equivalent classes. On the other hand, the concept of IFMOCs can be opted for proving the identifiability of a graph $\bar{G}$, for which the following theorem is given.

Assumption 5.1 (IFMOC:) *The true underlying functional model that generates the given process data-set belongs to an identifiable functional model class (IFMOC) with graph $\bar{G}$ such that $PA^{\bar{G}}(x_i)$ are the direct causes of the x_i, where $PA(.)$ represents the parent nodes.*

Theorem 5.1 *Assume that $P^{(x_i)}, x_i \in V$ is induced from an identifiable functional model class (IFMOC) with a graph $\bar{G}$. Then the same conditional/joint probability space can not be induced from the same IFMOC corresponding to a different graph $\bar{G}' \neq \bar{G}$.*

The proof of this theorem can be found in [111]. This theorem can be reformulated in the context of the causal inference such that if a given process data-set satisfies *Assumption 5.1*, then the unique graph $\bar{G}$ can be identified by the proposed strategy, which can be used to locate the root-cause fault variables.

Corollary 5.1 *If the IFMOC condition in Assumption 5.1 is satisfied, then the true causal DAG $\bar{G}$ from the joint distribution $P^{(X_i)}, i \in V$, can be identified.*

For a given process data-set, one can check if the *Assumption 5.1* is satisfied by using a statistical algorithm given in [111], which outputs the number of possible DAG. If the result of the algorithm is one ($\#DAG = 1$), then the time-series data satisfies *Assumption 5.1*. Furthermore, it is desirable for the causal graph $\bar{G}$ to be minimal, which means that its edges represent direct causal relationships instead of indirect/spurious ones. Therefore, the focus of this chapter is to first find a causal graph G among potential candidates x_i that represents the existing dependencies of their temporal conditional/joint probability distributions especially under a process fault, then such a graph G is further pruned to $\bar{G}$ by removing indirect/spurious edges.

For generality, x_s represents the hypothetical source variable and x_t represents the hypothetical target counterpart for investigating causal inference from x_s to x_t. If G representing causal structure of a given IFMOC is known, for the best case scenario, G is a DAG and the final goal is to extract its minimal sub-graph $\bar{G} \subset G$. This means that any edge $x_s \rightarrow x_t$ in $\bar{G}$ implies a direct causation from x_s to x_t (i.e. $x_s \in PA^{\bar{G}}(x_t)$) without presence of any intermediate variables.

In [56], it was shown that if the two variables x_s and x_t are dependent, then under certain assumption [56] [59]), both mutual information $I(x_s, x_t)$ and $TE_{x_s \rightarrow x_t}$ or $TE_{x_t \rightarrow x_s}$ can reveal

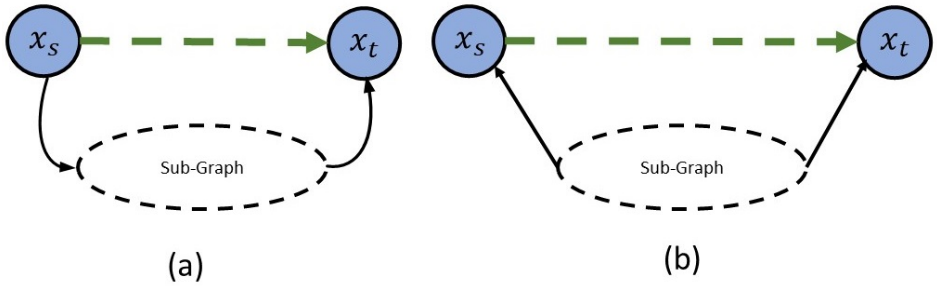


Figure 5.2: Illustration of spurious and indirect pathways $x_s \rightarrow x_t$. (a) indirect causal path, (b) spurious causal path.

the directional causal inference. Therefore, if the given process data-set satisfies *Assumption 5.1* and the general conditions mentioned in [56] [59] hold, according to Corollary 5.1, a unique causal graph $\bar{G}$ exists and TE/DTE can be used to identify it.

5.4 Direct and Indirect Causality

As discussed in the following, there exist two possibilities for which an edge detected by TE may not represent a true direct causal inference.

a) Fig. 5.2.(a) shows an *indirect* path from x_s to x_t represented by broken line, while the solid line represents true direct causal inference according to unknown IFMOC between variables. This indirect causal edge $x_s \rightarrow x_t$ is identified by utilizing TE only. This causality exists due to presence of one/some intermediate variables in the sub-graph that mediate from source to target. If the intermediate variables are known, one can apply a method (e.g. Direct TE as proposed in 5.5) to evaluate whether the edge is direct or not.

b) Fig. 5.2.(b) shows a *spurious* path from x_s to x_t. It should be noted that this spurious edge can be bi-directional or from x_t to x_s, but one of the three possibilities are presented in Fig. 5.2.(b).

It is worthwhile mentioning that the direct causal connection from x_s to x_t is a relative concept. If the only available measurements are x_s and x_t in Fig. 5.2 and all the confounding variables are not measured in the given data-set, the direct status of the causal relation $x_s \rightarrow x_t$ can not be fully investigated and as a result, the edge can be taken as a direct causal inference.

It is proven that under reasonable conditions, the TE can reveal all causal relations, e.g. direct, indirect and spurious, among nodes x_i and generate an initial graph G, knowing that the true causal graph representing the process IFMOC is $\bar{G} \subset G$. Also, if a sufficient number of intermediate variables are measured and available, then DTE can be used to prune the edges which do not represent direct causal inference, and accordingly it leads to recover graph $\bar{G}$ from G.

5.5 Proposed Symbolic Dynamic-Based Direct Transfer Entropy (SDDTE)

Given a set of potential candidate variables for root-cause fault diagnosis by following the steps shown in the Fig. 5.1, the *Shannon* entropies proposed in Eqs. (4.9) and (4.10) are utilized for a hypothetical source variable $x_s := x_a$ and a target variable $x_t := x_b$ to reformulate symbolic dynamic-based TE. Furthermore, in order to measure the strength of the information flow determined by proposed SDTE, the normalization approach proposed in [59] is applied to derive the symbolic dynamic-based normalized transfer entropy (SDNTE) as follows;

$$SDNTE_{x_a \longrightarrow x_b} = \frac{SDTE_{x_a \longrightarrow x_b}}{H_0 - SDH(x_b^{i+h\tau}|x_b^{l_1}, x_a^{l_2})}, \; H_0(x_b) = \log(x_b^{max} - x_b^{min}) \tag{5.2}$$

where the x_b^{max} and x_b^{min} denote the *maximum* and *minimum* values of time-series x_b. H_0 represents the maximal differential entropy of x_b with uniform distribution.

Eq. (5.2) is utilized to find a provisional (initial) directional causal graph G (causal map) indicating the information pathways while a malfunction exists in a process. The edges in G indicate either direct information pathways or the ones that exist through inter-connections of single/multiple intermediate variables (IVs). If a pathway is not direct, it is either spurious or indirect and must be pruned to avoid complications such as an increase in computational complexity and miss diagnose root-cause fault. In [116], partial TE is proposed to test whether the connection between two variables is spurious assuming all other variables as IVs. Considering all environmental variables as confounding contributors tremendously increases the computational complexity especially when kernel functions are utilized to estimate conditional probability functions (PDFs). On the other hand, DTE proposed in [59] has a similar formulation with partial TE, but it only considers a single or multiple particular IVs. Eq. (5.3) shows the DTE from x_a to x_b considering c number of IVs $z_j, j = 1, ..., c$.

$$DTE_{x_a \longrightarrow x_b}^{z_1,...z_c} = H(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) - H(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) \tag{5.3}$$

Conditional *Shannon* entropy $H(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c})$ is defined in Eq. (5.4) and represents the uncertainty about predictability of x_b according to the knowledge about past values of itself as well as all existing IVs $z_1, ..., z_c$.

$$\begin{aligned} &H(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) = \\ &-\sum_{i=1}^{N} p(x_b^{i+h\tau}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) \log p(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) \end{aligned} \tag{5.4}$$

Also, $H(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c})$ represents the uncertainty of predicting future values of x_b by knowing the past values of itself, all chosen IVs $z_1, ..., z_c$ and x_a. This entropy is defined in Eq. (5.5),

$$H(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) = \\ -\sum_{i=1}^{N} p(x_b^{i+h\tau}, x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) \log p(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}), \tag{5.5}$$

where $z_j^{v_1} = [z_j^i, z_j^{i-\tau}, ..., z_j^{i-(v_1-1)\tau}]$ is the embedding vector for the intermediate variable z_j.

DTE in Eq. (5.3) determines the amount of information about the future of x_b obtained from simultaneous observation of x_a and $z_j, j = 1, ..., c$ after discarding the information about future of x_b by only knowing the information of $z_j, j = 1, ..., c$. This means that if DTE is relatively non-zero, there is a direct information flow from x_a to x_b. In order to estimate a $(c + 2)$ dimensional joint-probability distribution function, one would try to apply *Gaussian* kernel fitting technique as indicated in [59]. However, this approach suffers from heavy computational complexity and curse of dimensionality, especially when the number of intermediate variables c increases. Therefore, the application of symbolic dynamic filtering (SDF) explained in section 4.4.2 is extended to estimate multi-dimensional joint/conditional PDFs (e.g. $p(x_b^{i+h\tau}, x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c})$ and $p(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c})$) in a time-efficient sense. This contribution enables utilization of DTE for real-time causality analysis application as discussed in the proposed autonomous root-cause fault diagnosis technique in this chapter. To this aim, a multi-dimensional joint *xD-Markov* machine is proposed as following which enables fast determination of the joint-PDF functions presented in Eq. (5.3) from a new SDF perspective,

Definition 5.1 (Multi-dimensional joint xD-Markov machine) *Let $\mathcal{A}^{\mathbf{x_a}}$ and $\mathcal{A}^{\mathbf{x_b}}$ be the PFSAs for the hypothetical source and target variables symbolic streams s^{x_a} and s^{x_b}, respectively. $\mathcal{A}^{\mathbf{z_1}}$, ... ,$\mathcal{A}^{\mathbf{z_c}}$ are the corresponding PFSAs for the selected intermediate variables. Then a multi-dimensional joint xD-Markov machine is defined as a $(c + 2)$-tuple $\mathcal{A}^{\mathbf{x_a x_b z_1 \ldots z_c \to x_b}} \triangleq \{Q^{\{x_a x_b z_1 \ldots z_c\}}, Q^{\{x_b z_1 \ldots z_c\}}, \Sigma^{x_a}, \Sigma^{x_b}, \Sigma^{z_1}, ..., \Sigma^{z_c}, \Delta^{x_a}, \Delta^{x_b}, \Delta^{z_1}, ..., \Delta^{z_c}, \Pi^{(z_1 \ldots z_c x_b)(z_1 \ldots z_c x_b)}, \tilde{\Pi}^{(x_a x_b z_1 \ldots z_c) x_b}\}$,*

1. *$\Sigma^{x_a}, \Sigma^{x_b}, \Sigma^{z_1}, ..., \Sigma^{z_c}$ are non-empty finite sets of alphabets belong to symbolic sequences $s^{x_a}, s^{x_b}, s^{z_1}, ..., s^{z_c}$, respectively.*

2. *$Q^{x_a}, Q^{x_b}, Q^{z_1}, ..., Q^{z_c}$ are finite sets for states of the corresponding symbol sequences.*

3. *$q_{r_3}^{\{x_b z_1 \ldots z_c\}} \in Q^{\{x_b z_1 \ldots z_c\}}$ represents a multi-dimensional joint state sequence similar to Def. 4.3-6 with $r_3 = 0, ..., |Q^{x_b}| \times |Q^{z_1}| \times .... |Q^{z_c}| - 1$. Similarly, $q_{r_5}^{\{x_a x_b z_1 \ldots z_c\}} \in Q^{\{x_a x_b z_1 \ldots z_c\}}$ is defined with $r_5 = 0, ..., |Q^{x_a}| \times |Q^{x_b}| \times |Q^{z_1}| \times .... |Q^{z_c}| - 1$.*

4. *$\mathbf{\Delta}^{(\cdot)} : Q^{(\cdot)} \times \Sigma^{(\cdot)} \to Q^{(\cdot)}$ is the general form of a state transition map which applies to every symbolic sequence involved in the calculation.*

5. *$\mathbf{\Pi}^{(\mathbf{x_b z_1 \ldots z_c})(\mathbf{x_b z_1 \ldots z_c})}$ is a square matrix of size $(|Q^{\{x_b z_1 \ldots z_c\}}| = |Q^{x_b}| \times |Q^{z_1}| \times ... \times |Q^{z_c}|)$; where $\pi_{r_3 r_4}^{(x_b z_1 \ldots z_c)(x_b z_1 \ldots z_c)}$ is probability of moving from multi-dimensional joint state $q_{r_3}^{\{x_b z_1 \ldots z_c\}}$ at n^{th} epoch to $q_{r_4}^{\{x_b z_1 \ldots z_c\}}$ at $(n + 1)^{th}$ epoch for $r_3, r_4 = 0, ..., |Q^{\{x_b z_1 \ldots z_c\}}| - 1$.*

6. $\tilde{\mathbf{\Pi}}^{(\mathbf{x_b z_1 \dots z_c})\mathbf{x_b}}$ *is the output symbol emission matrix of size* $(|Q^{\{x_b z_1 \dots z_c\}}|\times|\Sigma^{x_b}|)$*; where* $\tilde{\pi}^{(x_b z_1 \dots z_c)x_b}_{r_3 k}$ *is probability of observing* $\sigma^{x_b}_k \in \Sigma^{x_b}$ *as the* $(n+1)^{th}$ *symbol in the sequence* s^{x_b}*, while making a transition from the multi-dimensional joint state sequence* $q^{\{x_b z_1 \dots z_c\}}$ *at epoch* n^{th}*.*

7. $\mathbf{\Pi}^{(\mathbf{x_a x_b z_1 \dots z_c})(\mathbf{x_a x_b z_1 \dots z_c})}$ *is a square matrix of size* $\big(|Q^{\{x_a x_b z_1 \dots z_c\}}|= |Q^{x_a}|\times|Q^{x_b}|\times|Q^{z_1}|\times \dots \times |Q^{z_c}|\big)$*, where* $\pi^{(x_a x_b z_1 \dots z_c)(x_a x_b z_1 \dots z_c)}_{r_5 r_6}$ *is the probability of moving from multi-dimensional joint state* $q^{\{x_a x_b z_1 \dots z_c\}}_{r_5}$ *at epoch* n^{th} *to* $q^{\{x_a x_b z_1 \dots z_c\}}_{r_6}$ *at* $(n+1)^{th}$ *for* $r_5, r_6 = 0, \dots, |Q^{\{x_a x_b z_1 \dots z_c\}}|-1$*.*

8. $\tilde{\mathbf{\Pi}}^{(\mathbf{x_a x_b z_1 \dots z_c})\mathbf{x_b}}$ *is the output symbol emission matrix of size* $(|Q^{\{x_a x_b z_1 \dots z_c\}}|\times|\Sigma^{x_b}|)$*; where* $\tilde{\pi}^{(x_a x_b z_1 \dots z_c)x_b}_{r_5 k}$ *is the probability of observing* $\sigma_k \in \Sigma^{x_b}$ *as the* $(n+1)^{th}$ *symbol in the sequence* s^{x_b}*, while making a transition from the multi-dimensional joint state sequence* $q^{\{x_a x_b z_1 \dots z_c\}}_{r_5}$ *at epoch* n^{th}*. This multi-dimensional joint morph emission matrix is shown in Fig. 4.3.*

The concept of all transition and emitting matrices presented above are pictorially illustrated in Fig. 4.3. Def. 5.1, explains the basis of the symbolic-dynamic approach for fast estimation of the joint and conditional multi-dimensional PDFs which are required in the calculation of DTE. As a result, the multi-dimensional conditional probabilities exist in Eqs. (5.4) and (5.5) can be calculated using *morph* matrix (see Def. 5.1, step 6 and 8) as following,

$$\begin{aligned}
&\sum_{i=1}^{N} p(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, \dots, z_c^{v_c}) \approx \sum_{r_3=0}^{|\Sigma^{x_b}|-1} \sum_{k=0}^{|Q^{\{x_b z_1 \cdots z_c\}}|-1} \tilde{\pi}^{(x_b z_1 \dots z_c)x_b}_{r_3 k} \\
&\sum_{i=1}^{N} p(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, \dots, z_c^{v_c}) \approx \sum_{r_5=0}^{|\Sigma^{x_b}|-1} \sum_{k=0}^{|Q^{\{x_a x_b z_1 \cdots z_c\}}|-1} \tilde{\pi}^{(x_a x_b z_1 \dots z_c)x_b}_{r_5 k}.
\end{aligned} \tag{5.6}$$

With regards to the Def. 5.1, the joint probability $p(x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, \dots, z_c^{v_c})$ is approximated by $p(q^{\{x_b z_1 \dots z_c\}}_{r_3})$ calculated as the left eigenvector of unit eigenvalue of the state transition matrix $\mathbf{\Pi}^{(\mathbf{x_b z_1 \dots z_c})(\mathbf{x_b z_1 \dots z_c})}$. Similarly, the other joint PDF is symbolically approximated as $p(x_b^{l_1}, x_a^{l_2}, z_1^{v_1}, z_2^{v_2}, \dots, z_c^{v_c}) \approx p(q^{\{x_a x_b z_1 \dots z_c\}}_{r_5})$. Moreover, according to *Bayes'* rule $p(A, B) = p(B)p(A|B)$ and Eq. (5.6), the symbolic dynamic entropies are derived as following,

$$\begin{aligned}
&SDH(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, \dots, z_c^{v_c}) = \\
&\sum_{k=0}^{|\Sigma^{x_b}|-1} \sum_{r_3=0}^{|Q^{\{x_b z_1 \cdots z_c\}}|-1} p(q^{\{x_b z_1 \dots z_c\}}_{r_3}) \tilde{\pi}^{(x_b z_1 \dots z_c)x_b}_{r_3 k} \log \tilde{\pi}^{(x_b z_1 \dots z_c)x_b}_{r_3 k} \\
&SDH(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, \dots, z_c^{v_c}) = \\
&\sum_{k=0}^{|\Sigma^{x_b}|-1} \sum_{r_5=0}^{|Q^{\{x_a x_b z_1 \cdots z_c\}}|-1} p(q^{\{x_a x_b z_1 \dots z_c\}}_{r_5}) \tilde{\pi}^{(x_a x_b z_1 \dots z_c)x_b}_{r_5 k} \log \tilde{\pi}^{(x_a x_b z_1 \dots z_c)x_b}_{r_5 k}
\end{aligned} \tag{5.7}$$

Eq. (5.7) presents the symbolic dynamic representation of the *Shannon* entropies given in Eq. (5.3). Thus, by substituting the SDHs in Eq. (5.3), the proposed symbolic dynamic-based transfer entropy $SDDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c}$ is given as follows,

$$\begin{aligned} & SDDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c} = \\ & SDH(x_b^{i+h\tau}|x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) - SDH(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c}) \end{aligned} \tag{5.8}$$

Therefore, in order to measure the strength of the direct causality from x_a to x_b considering all intermediate variables $z_1, ..., z_c$, the same normalization procedure similar to Eq. (5.2) is utilized and the symbolic dynamic-based normalized direct transfer entropy $SDNDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c}$ is derived as follows,

$$SDNDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c} = \frac{SDDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c}}{SDH({x_b}^{i+h\tau}|{x_b}_i^{l_1}) - SDH(x_b^{i+h\tau}|x_a^{l_2}, x_b^{l_1}, z_1^{v_1}, z_2^{v_2}, ..., z_c^{v_c})} \tag{5.9}$$

The numerator in the above equation represents the $SDDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c}$ and the denominator is the total causality from both x_a and intermediate variables $z_1, ..., z_c$ to x_b. Therefore, $SDNDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c}$ intuitively determines the percentage of direct causality in the total causality from both x_a and $z_1, ..., z_c$ to x_b. For the ease of implementing the proposed symbolic-dynamic based SDNTE in Eq. (5.2) and SDNDTE in Eq. (5.9), the frequency counting formulas for all the required components are summarized in Table 5.1.

5.5.1 Computation Complexity of Proposed $SDNDTE_{x_a \longrightarrow x_b}^{z_1,...,z_c}$

One of the advantages of the proposed $SDNDTE$ in Eq. (5.9) is the computational efficiency in comparison with the traditional method based on estimating the joint-conditional PDFs shown in Eqs. (5.4) and (5.5) using kernel functions [59] [57]. The computational order of a c_d dimensional joint PDF is $O(N^2 c_d^2)$ using Fukunaga method [61]. The maximum dimension of the joint PDF in Eq. (5.3) is $c_d = l_1 + l_2 + v_1 + ... + v_c + 1$ which is the sum of the dimensions of the embedding vectors. Therefore, the total computational complexity of calculating TE using kernel function is $O(N^2(l_1 + l_2 + v_1 + ... + v_c)^2)$.

The computation complexity of the proposed method in this chapter is due to two main factors in two steps of the proposed algorithm. The first step is generation of the symbolic sequences out of the m time-series as shown in Fig. 4.2 which can be done by complexity order of $O(mN)$. In the second step, according to Def. 5.1, calculation of the proposed $SDNDTE$ is based on the frequency counting approach. In Table 5.1, the formula for calculating each component exist in Eq. (5.9) (e.g. state transition matrices and morph emission matrices) are provided. It is proposed to deploy frequency counting function $N(.,.)$ using index-based linear search in which all the elements of Π and $\tilde{\Pi}$ are determined within one loop with size of N (i.e. length of the symbolic sequences). Therefore, the calculation complexity of all components in proposed $SDNDTE$ is $O(N)$. In addition, eigenvector decomposition required to calculate $p(q_j^{x_a}, q_j^{x_b}, q_{i_1}^{z_1}, ..., q_{i_n}^{z_c})$ has theoretical calculation complexity of $O(|Q^{\{x_a x_b z_1 ... z_c\}}|^{2.376})$ [117]. According to the summations exist in Eqs.

Table 5.1: The frequency counting equations for all of the symbolic-dynamic probability terms required for proposed SDNTE Eq. (5.2) and SDNDTE Eq. (5.9). Note that $N(.)$ represents the number of occurrences

Causality terms	Frequency counting formula
$p(q_j^{x_b})$	$\frac{N(q_j^{x_b})}{\sum_{i=0}^{\lvert Q^{x_b}\rvert} N(q_i^{x_b})}$
$\pi_{ij}^{x_a x_a}$	$\frac{N(q_i^{x_a}, q_j^{x_a})}{\sum_{k=0}^{\lvert Q^{x_a}\rvert -1} N(q_i^{x_a}, q_k^{x_a})}$
$\tilde{\pi}_{jk}^{x_b x_b}$	$\frac{N(q_j^{x_b}, \sigma_k^{x_b})}{N(q_j^{x_b})}$
$\pi_{r_1 r_2}^{(x_a x_b)(x_a x_b)}$	$\frac{N(q_{r_2}^{\{x_a x_b\}}, q_{r_1}^{\{x_a x_b\}})}{\sum_{r=0}^{\lvert Q^{\{x_a, x_b\}}\rvert -1} N(q_{r_2}^{\{x_a x_b\}}, q_r^{\{x_a x_b\}})}$
$P(q_i^{x_a}, q_j^{x_b})$	*is the left eigenvector of* *unit eigenvalue of* $\Pi^{(x_a x_b)(x_a x_b)}$
$\tilde{\pi}_{r_1 k}^{(x_a x_b) x_b}$	$\frac{N(q_{r_1}^{\{x_a x_b\}}, \sigma_k^{x_b})}{N(q_{r_1}^{\{x_a x_b\}})}$
$\pi_{r_3 r_4}^{(x_b z_1 \ldots z_c)(x_b z_1 \ldots z_c)}$	$\frac{N(q_{r_4}^{\{x_b z_1 \ldots z_c\}}, q_{r_3}^{\{x_b z_1 \ldots z_c\}})}{\sum_{r=0}^{\lvert Q^{\{x_b, z_1, \ldots, z_c\}}\rvert -1} N(q_{r_4}^{\{x_b z_1 \ldots z_c\}}, q_r^{\{x_b z_1 \ldots z_c\}})}$
$p(q_j^{x_b}, q_{i_1}^{z_1}, ..., q_{i_n}^{z_c})$	*is the left eigenvector of* *unit eigenvalue of* $\Pi^{(x_b z_1 \ldots z_c)(x_b z_1 \ldots z_c)}$
$\pi_{r_5 r_6}^{(x_a x_b z_1 \ldots z_c)(x_a x_b z_1 \ldots z_c)}$	$\frac{N(q_{r_5}^{\{x_a x_b z_1 \ldots z_c\}}, q_{r_6}^{\{x_a x_b z_1 \ldots z_c\}})}{\sum_{r=0}^{\lvert Q^{\{x_a, x_b, z_1, z_c\}}\rvert -1} N(q_{r_5}^{\{x_a x_b z_1 \ldots z_c\}}, q_r^{\{x_a x_b z_1 \ldots z_c\}})}$
$p(q_s^{x_a}, q_j^{x_b}, q_{i_1}^{z_1}, ..., q_{i_n}^{z_c})$	*is the left eigenvector of* *unit eigenvalue of* $\Pi^{(x_a x_b z_1 \ldots z_c)(x_a x_b z_1 \ldots z_c)}$
$\tilde{\pi}_{s j i_1 \ldots i_q k}^{(x_a x_b z_1 \ldots z_c) x_b}$	$\frac{N(q_{r_5}^{\{x_a x_b z_1 \ldots z_c\}}, \sigma_k^{x_b})}{N(q_{r_5}^{\{x_a x_b z_1 \ldots z_c\}})}$

(5.7) and (5.9), the total complexity order of calculating $SDNDTE_{x_a \xrightarrow{} x_b}^{z_1,...,z_c}$ is $O(mN + N + |Q^{\{x_a x_b z_1 \ldots z_c\}}|^{2.376} + |\Sigma^{x_b}||Q^{\{x_a x_b z_1 \ldots z_c\}}| + |\Sigma^{x_b}||Q^{\{x_b z_1 \ldots z_c\}}| + |\Sigma^{x_b}||Q^{x_b}| + |\Sigma^{x_b}||Q^{x_b}||Q^{x_b}|)$ which is equal to $O((m+1)N + |Q^{\{x_a x_b z_1 \ldots z_c\}}|^{2.376})$.

5.6 Proposed Autonomous Framework for Root-Cause Fault Diagnosis

This section explains the proposed autonomous framework for finding the root-cause fault in a process. It is important to know that this proposed framework is a general paradigm for root-cause diagnosis and it is not limited to the application of proposed SDNTE and SDNDTE for causality analysis. Hence, other techniques for causality analysis (e.g. Granger causality, constraint-based causality analysis [118], time-warping [50], etc.) can also be fit into this proposed framework. In this chapter, symbolic-dynamic TEs are considered as the main tool to determine whether there is a causal relationship between two variables and validate direct causal inference. Therefore, by applying the proposed SDNTE in Eq. (5.2) between every pair of variables, a directed graph is derived indicating the causality map between potential candidates for being root-cause(s) of the detected fault. In this graph, the vertices represent process measurements and the edges represent causal information pathways between them.

According to the intuition behind definition of transfer entropy, if $TE_{x_s \rightarrow x_t}$ is relatively significant, it only indicates that there is possibly a direct (immediate) pathway $x_s \rightarrow x_t$. In other words, there is no guarantee that this detected information pathway is direct and it is not due to the presence of single/multiple IVs. To this end, the proposed symbolic dynamic-based normalized direct transfer entropy $SDNDTE_{x_s \xrightarrow{} x_t}^{z_1,...,z_c}$ in Eq. (5.9) is proposed as a viable technique to test whether the detected pathway $x_s \rightarrow x_t$ is a direct pathway or it is possibly induced by the presence of one/some IV(s).

It should be noted that the potential intermediate variables $z_1, ..., z_c$ for determining $SDNDTE_{x_s \xrightarrow{} x_t}^{z_1,...,z_c}$ must be efficiently selected since the calculation complexity of SDNDTE depends on the number of selected intermediate variables (c). Hence, only the true intermediate variables must be considered if the relatively fast determination of accurate results is the final goal. For this purpose, explicit definitions of two types of IVs (e.g. *Immediate* IV in Def. 5.2 and *Source* IV in Def. 5.3) are introduced. Finding intermediate variables between x_s and x_t is conventionally done manually by an expert or by looking at the topology map of the process which is a challenge for autonomous real-time applications. Therefore, this paper proposes a systematic paradigm to autonomously determine the intermediate variables and further determine the root-cause fault without any need for intervention of an expert.

5.6.1 Proposed Algorithms to Find Immediate and Source Intermediate Variables (IVs)

Assume that a process with n variables is given and a FD method (e.g. Kernel PCA recalled in section (4.3)) is applied for detecting process fault. Upon detection of a fault in the process,

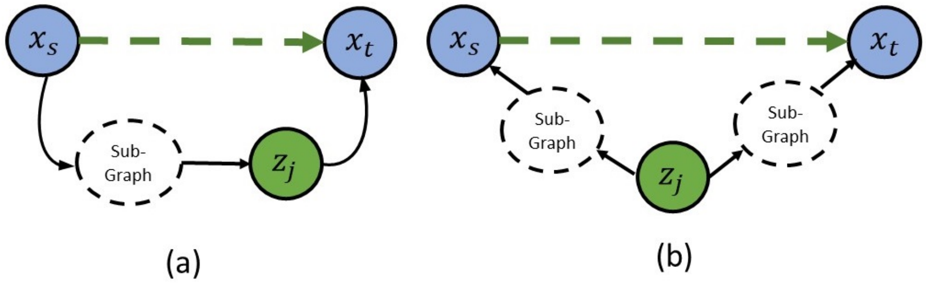


Figure 5.3: Illustration of spurious and indirect pathways $x_s \rightarrow x_t$ through either presence of an immediate or source intermediate variable. (a) Immediate intermediate variable (IIV) case, (b) Source intermediate variable (SIV) case.

a conventional diagnosis technique (e.g. accumulative rate contribution (ACRC) [2]) is used to choose only $m \leq n$ faulty variables as potential root-cause(s). The first step for finding root-cause of the detected fault is to apply SDNTE in Eq. (5.2) to determine an initial causal directed graph representing information pathways. The second step is to efficiently utilize SDNDTE proposed in Eq. (5.9) to prune indirect and spurious pathways. With regards to the underlying intuition of the SDNDTE, if the pathway $x_s \rightarrow x_t$ is not a direct pathway, there must be a single or multiple IV(s) inducing this indirect/spurious pathway detected by SDNTE.

Definition 5.2 (Immediate Intermediate Variable (IIV)) *When there exists an indirect path ($x_s \rightarrow x_t$ as shown in Fig. 5.3(a) by a broken green arrow) and at the same time there exist one/multiple other direct paths connecting x_s to x_t in the sense that the path is unidirectional and through a set of intermediate variable(s) (IVs), the last variable in such path(s) that is immediately before x_t is defined as the immediate intermediate variable (IIV).*

Definition 5.3 (Source Intermediate Variable (SIV)) *Under the circumstance that a spurious path exists between x_s and x_t (as shown in Fig. 5.3(b) by broken green arrow) in the sense that x_s and x_t are affected simultaneously by certain common source (parent) variable(s) excluding the x_s, these variable(s) are defined as source intermediate variables (SIVs).*

Fig. 5.3(a) indicates a pictorial explanation for the case that IIV results in the presence of an indirect edge $x_s \rightarrow x_t$ denoted by broken green line. Although along a path from x_s to x_t several IVs may exist, the variable right before x_t in the path is the only one required for calculation of SDNDTE. The intuition behind this assumption is that if $x_s \rightarrow x_t$ is not directly due to the transmission of information from x_s to x_t through other IVs shown in Fig. 5.3, the complete transmitted information through that path exists in the last variable immediate adjacent to x_t. Therefore, it suffices to only consider this variable, which we define as IIV (z_j in Fig. 5.3). It should be noted that not only incorporating other variables in addition to IIV does not affect the result of DTE calculation, it may also make the calculation of SDNDTE inaccurate.

Before presenting the proposed algorithms to find IIV and SIV, two objectives of this subsection are recapitulated. First, the developed algorithms must not select redundant IVs that do not affect the result of $SDNDTE^{z_1,\dots,z_c}_{x_s\longrightarrow x_t}$. This objective guarantees to identify only the effective IVs, which as a result minimizes the calculation complexity . Second, develop algorithm(s) to autonomously find all c existing IIVs and SIVs required for calculation of $SDNDTE^{z_1,\dots,z_c}_{x_s\longrightarrow x_t}$ in Eq. (5.9).

Algorithm 3 is proposed to efficiently find IIVs z_j between x_s and x_t using the idea of depth first search (DFS) method [119]. There are different ways to validate the correctness of an algorithm such as induction, case analytics, and contradiction. Hereby we investigate correctness of the proposed *Algorithm* 3 by contradiction.

Algorithm 3 Find immediate intermediate variables (IIVs) between x_s and x_t

```
stack → $s$, path → $p$, vertex → $v$, Neighbours → $Neigh$, node → $n$, graph → $G$
Inputs:G, $x_s$, $x_t$
$s \leftarrow x_s$
$V_{IIV} \leftarrow empty$
loop(1): $s \neq empty$
$p_{IIV} \leftarrow s[end]$, and remove it from $s$
$v \leftarrow$ last node of $p_{IIV}$
$Neigh = G[v]$
loop(2): $n \in Neigh$
if $n \notin p_{IIV}$ then:
    $p^{new}_{IIV} \leftarrow p_{IIV}$
    $p^{new}_{IIV} \leftarrow p^{new}_{IIV} + n$
    $s \leftarrow s + p^{new}_{IIV}$
    if $\{n = x_t\}$ & $\{length(p^{new}_{IIV}) \geq 3\}$ then:
        if $\{p^{new}_{IIV}[length(p^{new}_{IIV}) - 2] \notin V_{IIV}\}$ then:
            $V_{IIV} \leftarrow p^{new}_{IIV}[length(p^{new}_{IIV}) - 2]$
            go to loop(2).
go to loop(1).
Output:$V_{IIV}$
```

Assume that there are m_1 potential confounding variables exist to be incorporated in validation of the information pathway $x_s \rightarrow x_t$. After implementing *Algorithm* 3, $n_1 \leq m_1$ of the potential variables are chosen as IIV. According to the proof of correctness by contradiction, we assume that there is another variable x_r that is indeed an intermediate (confounding) variable and is missed by the proposed Algorithms. With respect to Def 5.2, it means that there is a path from x_s to x_t passing through x_r such that none of the variables in this path exist in those chosen $n_1^{IIV} \leq n_1$ variables. Hence, with regards to the intuition explained in Def 5.2, there must be one path from x_s to x_t that is missed by *Algorithm* 3. On the other hand, the underlying basis of the proposed *Algorithm* 3 is depth first search (DFS) method. Therefore, this assumption leads to the fact that the DFS method failed to find all possible distinct paths from x_s to x_t, which is not possible.

As depicted in Fig. 5.3(b), since z_j is the common source of information to both x_s and x_t, there is a similar piece of information present in both x_s and x_t transmitted from z_j which causes this spurious pathway. The proposed *Algorithm* 4 efficiently finds the SIVs, in which all of the variables except the source, target and those ones that are already chosen as intermediate variables are topologically tested to be possibly SIVs according to the initial causal graph G. As can be seen in Fig. 5.3(b), if the path from z_j to either x_s (or x_t) passes through x_t (or x_s), the path is ignored. The intuition behind the proposed *Algorithm* 4 is to find those variables that might induce a similar piece of information into x_s and x_t which lead to generation of a spurious path (i.e. edge in the graph $x_s \to x_t$). In *Algorithm* 4, depth first search (DFS) is used to find all possible path between two vertices. The proposed algorithm consists of one procedure as the main paradigm (i.e. line 1-11) and one procedure as a helper function (line 9). The main part of *Algorithm* 4 indicates the steps of validating each potential common source variable. On the other hand, a helper function is proposed to validate the suggested intuition in this research to efficiently find true SIVs.

Algorithm 4 Find source intermediate variables (SIVs) between x_s and x_t

```
 1: path $\to P$, graph $\to G$
 2: **Inputs:** $G$, *visited*, $x_s$, $x_t$
 3: $V_{SIV} \leftarrow empty$
 4: ***loop***: $k = 0, ..., m-1$, ($m \leftarrow$ number of vertices in $G$)
 5: $dummy_A \leftarrow visited + x_t$
 6: $P_{x_s}^{all} \leftarrow$ all possible path from $k$ to $x_s$ using DFS Algorithm, when $visited \leftarrow dummy_A$
 7: $dummy_B \leftarrow visited + x_s$
 8: $P_{x_t}^{all} \leftarrow$ all possible path from $k$ to $x_t$ using DFS Algorithm, when $visited \leftarrow dummy_B$
 9: ***Helper Function*** *(Algorithm 9)*:Inputs$\Rightarrow P_{x_s}^{all}, P_{x_t}^{all}$, $Output = V_{SIV}^{potential}$
10: Append the elements of $V_{SIV}^{potential}$ to $V_{SIV}$ if they do not already exist there
11: **Outputs:** $V_{SIV}$
```

Similar to the proposed algorithm for finding the IIVs, *Algorithm* 4 can find all necessary source intermediate variables needed for determination of $SDNDTE_{x_s \xrightarrow{z_1,...,z_c} x_t}$. The following explains the correctness for the proposed *Algorithm* 4.

Assume that there are m_1 potential confounding variables to be incorporated in validation of the information pathway $x_s \longrightarrow x_t$. After implementing *Algorithm* 4, $n_2 \leq m_1$ of the potential variables are chosen as SIVs. The contradicting assumption is that there is an SIV x_r that is not found by *Algorithm* 4. According to the intuition behind Def. 5.3, this assumption infers that there must be two distinct paths, one from x_r to x_s, and one from x_r to x_t. Moreover, DFS is the basis of the *Algorithm* 4 and it is deployed to find all possible paths from x_r to x_s and x_t. This leads to the fact that DFS failed to find all possible paths, which is a contradiction.

5.6.2 Proposed Autonomous Technique for Root-Cause Fault Diagnosis

In Algorithms 3 and 4, the proposed idea for finding the IVs are presented, as one of the core components of the proposed autonomous paradigm for root-cause fault diagnosis. Each edge in the extracted directed graph G from a hypothetical source variable x_s to a hypothetical target variable x_t is required to be validated by $SDNDTE_{x_s \xrightarrow{} x_t}^{z_1,...,z_c}$. This is where the proposed Algorithms 3 and 4 play their prominent roles to efficiently determine the IV(s) $z_1, ..., z_c$. After pruning the indirect edges of G and generate $\bar{G}$, this graph is efficient enough to be utilized for finding the root-cause fault variable(s). The proposed idea for autonomously determining the root-cause vertices in the pruned causal graph is developed in *Algorithm* 5.

Remark 5.1 *One assumption in this algorithm is that if there exists only one fault in the process, the source of the fault information in a directed graph must have a path to at least one node of the graph for the sake of detectability and diagnosability.*

The idea behind the fault diagnosis step explained in step 2 mentioned in section 5.1 is that any process variable that is affected by the existing fault is successfully diagnosed and has a representing node in the graph G for the *Algorithm* 5. As a particular case, if a process is subjected to a fault that only affects one process variable, and that orphan variable does not have any causal connections to any other variables, *Algorithm* 5 will output that variable as a potential root-cause fault.

Remark 5.2 *The proposed strategy for root-cause fault diagnosis is not limited to the single process fault occurrence at a time. If multiple faults occur simultaneously, they introduce more sources of information in the pruned graph $\bar{G}$. Therefore, Algorithm 5 reports all of them as potential sources of detected fault. At that point, a domain expert or additional analysis will be required to distinguish between different fault scenarios.*

Algorithm 6 summarizes the step-by-step actions proposed in this chapter to autonomously find the root-cause fault in the process. The final output of *Algorithm* 6 is the measurement variable(s) that are identified as source of the detected fault.

5.7 Simulation Results

In the first part of this section, a synthetic numeric process is simulated and a hidden fault is introduced to some variables. The IFMOC shown in Eq. (5.10) is defined such that satisfies *Assumption 5.1* stated in section 5.3. Therefore, the proposed strategy can be applied to identify the graph $\bar{G}$ representing the Eq. (5.10). In this numerical example, fault detection step (e.g. Kernel PCA) and faulty variable diagnosis (e.g. accumulative rate contribution (ACRC) index) counterpart are skipped and the only goal is to evaluate the efficiency and applicability of the proposed SDNTE, SDNDTE and autonomous algorithms for deriving causal graph and

Algorithm 5 Proposed autonomous procedure to find the root-cause fault using the pruned causality graph

```
procedure (FINDING THE ROOT-CAUSE FAULT VARIABLE(S)):
    $List_{var} \leftarrow$ list of all vertices in the graph
    $Root_{causes} \leftarrow List_{var}$
    loop(1): $V_{candidate} \in List_{var}$
    if $V_{candidate} \in Root_{causes}$ then
        if $G[V_{candidate}]$ is not empty then:
            $dummy_1 \leftarrow$ DFS Helper function with inputs: $G$, $V_{candidate}$
            loop(2): $var \in dummy_1$
            $dummy_2 \leftarrow$ DFS Helper function with inputs: $G$, $var$
            if ($G[V_{candidate}]$ is empty) & ( $var$ is in $Root_{causes}$) then:
                remove $var$ from $Root_{causes}$
            else if ($V_{candidate}$ is not in $dummy_2$) & ($var$ is in $Root_{causes}$) then:
                remove $var$ from $Root_{causes}$
    Output:$Root_{causes}$
————————————————————————————————
procedure (DFS HELPER FUNCTION TO DETERMINE ALL REACHABLE NODES OF A GRAPH STARTS FROM $x_s$):
    DFS Inputs: $G$, $x_s$
    $s \leftarrow x_s$
    $visited \leftarrow empty$
    loop(1): $s \neq empty$
    $node \leftarrow$ last element of the $s$
    Remove the last element of the $s$
    loop(2): $n \in Neighbours$
    if $n \notin visited$ then:
        $visited \leftarrow visited + n$ (append $node$ to the $visited$)
        $s \leftarrow s + n$ (append $node$ to end of the $s$)
    Output:$visited$
```

Algorithm 6 Summery of the proposed strategy for autonomous root-cause fault diagnosis

```
Apply a fault detection method (e.g. KPCA or PCA) and detect fault(s),
Select root-cause fault candidates $x_i, i = 1, ..., m$ using a FD method (e.g. ACRC),
Generate a directed causal graph $G$ amongst $m$ candidates using proposed SDNTE method
  in Eq. (5.2),
—**Prune indirect and /or spurious edges in $G$**—
***loop(3)***: $i = 0, ..., m-1$,
$x_s \leftarrow G[i]$
    ***loop(4)***: $k \in G[x_s]$ (find all neighbour vertices to $x_s$)
$x_t \leftarrow G[x_s][k]$
$V_{Inter} \leftarrow empty$ (Initialize intermediate variables which will store both IIVs and SIVs)
————————$Find\ IIVs$————————-
$V_{Inter} \leftarrow V_{Inter} + Output$ of *Algorithm* 3 with inputs: $\{G, x_s, x_t\}$
———————— $Find\ SIVs$ ————————
$V_{Inter} \leftarrow V_{Inter} + Output$ of *Algorithm* 4 with inputs: $\{G, V_{Inter}, x_s, x_t\}$
After finding all possible intermediate variable between $x_s$ and $x_t$, conduct SDNDTE as
  indicated in Eq. (5.9) and validate it if the path way $x_s \rightarrow x_t$ is indirect/spurious or
  direct.
**if** $x_s \rightarrow x_t$ is spurious ($SDNDTE^{InterVar}_{x_s \rightarrow x_t} < UCL$) **then**: ($UCL = 0.1$ in this study)
    remove edge $x_s \rightarrow x_t$ from $G$
**go to** *loop(3)*.
**Output:**Pruned graph with direct information connection
–**Locate the root-cause fault using pruned $G$**–
Apply *Algorithm* 5 with input of pruned graph $G$.
```

consequently finding root-cause fault variables. In the second part, Tennessee Eastman Process (TEP) is considered to show the usefulness as well as the performance of the proposed general framework for industrial fault scenarios.

5.7.1 Numerical Example

Assume an additive noise model (ANM) including seven non-linear continuous random variables $x_i, i = 0, ..., 6$ as follows;

$$\begin{cases} x_0(i) &= 1 - 1.5e^{-0.2x_0(i-1)+2} + sin(x_1(i-2)) + 0.1x_5(i-1) * x_5(i-2) + u_0(i-1) \\ x_1(i) &= 2 + \dfrac{0.2x_0(i-1)}{12 - x_0(i-1)} - 0.3x_1(i-1) + u_1(i-1) \\ x_2(i) &= -0.5x_1(i-1) + 0.2x_2(i-1)x_5(i-1) + \sqrt{|x_3(i-1) - 2|} + u_2(i-1) \\ x_3(i) &= 3 + 0.1x_4(i-1)^2 - 0.2x_6(i-1) + \sqrt{|x_6(i-1)|} + u_3(i-1) \\ x_4(i) &= 1 - 0.05x_4(i-1)^2 - 0.4x_3(i-1)sin(x_4(i-2)) + u_4(i-1) \\ x_5(i) &= \mu_0(i-1) + 0.2\mu_0(i-2) + F(i-1) + u_5(i-1) \\ x_6(i) &= \mu_1(i-1) - 0.3\mu_1(i-2) + 0.2F(i-1)sin(F(i-1)) + u_6(i-1), \end{cases} \tag{5.10}$$

where $\mu_i, i = 0, 1$ are the two i.i.d exogenous inputs with zero mean and unit variance. $u_i \sim N(0, 0.01), i = 0, ..., 6$ are the independent measurement disturbances. It should be noted that the initial condition of zero is considered for all variables. According to the identifiability conditions in [112];

1- This process is index-based and causal.

2- f_is are non-linear which insures the causal identifiability among the measurements [120]. g_i is also unity and invertible.

3- $u_i \sim N(0, 0.1), i = 0, ..., 6$ are additive and mutually independent.

4- According to the SEM shown in Eq. (5.10), each variable x_i is independently defined from those ones that are not function or sub function of x_i, conditioned on the parent(s) of x_i.

5- The additive u_i corresponding to x_i is defined such that they are mutually and conditionally independent from $PA(x_i)$.

In Eq. (5.10), F is a random stationary independent fault with an average mean of zero and variance of 2 that is added to x_5 and x_6 and it is considered unknown from the measurements. Since the added fault is immeasurable in this example and the root-cause of the fault is a relative concept with respect to the available measurements, as a result of utilizing the proposed autonomous framework, x_5 and x_6 must be diagnosed as the true root-causes of the fault.

The process is simulated for 5000 samples and the fault is added at 4000^{th} sample. The first 1000 sample of the simulation is discarded to ensure the stationarity of the time series.

Fig. 5.4 shows 1000 samples of all seven variables. The assumption is that the time when the fault is introduced into the process is known upon its detection and all of the variables are considered as candidates of root-cause fault. Therefore, the goal is to utilize the proposed SDNTE and SDNDTE as well as autonomous algorithms to, first, derive an initial causal map, second, discard spurious and indirect edges and finally determine the root-cause(s) of the fault. To this aim, the parameters required for symbolizing the time series should be determined according to the steps proposed by authors in [20]. It should be mentioned that the number of states is reduced by applying the state merging technique mentioned in [99] to neglect those unlikely states in the estimation procedure to decrease the calculation memory/complexity the naive estimators included in Table 5.1. These parameters are determined and derived for the first 2000 samples in normal condition and the results are presented in Table 5.2. Moreover, the number of samples required for conducting causality analysis is determined as $N^{sd} = 1000$.

Table 5.2: Selected parameters for the state machine construction of the synthetic numerical example.

Index	ϵ_h	$\lvert\Sigma\rvert$	D	$\lvert Q\rvert$
X0	0.10	3	2	9
X1	0.10	5	2	23
X2	0.10	5	2	18
X3	0.10	3	2	9
X4	0.10	4	2	16
X5	0.10	3	2	9
X6	0.10	3	2	9

After finding the tuning parameters and symbolizing the time series, the next step is to apply the proposed SDNTE in Eq. (5.2) amongst all pairs of variables to generate an initial causal graph. All the required component to calculate the SDNTE is presented in Table 5.1. As can be seen in Table 5.4, the calculated SDNTE between each pair of variables is presented. In order to generate the initial causal graph indicating the information pathways, each connection indicated in Table 5.4 is considered as a pathway (i.e. graph edge) if its value is greater than a significant level of 10 percent. In Fig. 5.5, the initial graph is indicated on the top left corner in which edges represent significant pathways in Table 5.4. According to *Algorithm* 6 which summarizes the proposed autonomous framework, the next step is to check whether the connection between variables (i.e. each edge in the initial causal graph) is direct. Then for each validation of a connection, the *Algorithms* 3 and 4 are utilized to find the IIVs and SIVs, respectively. After autonomously and efficiently finding intermediate variables z_j between a source variable x_s to a target variable x_t, $SDNDTE_{x_s \to x_t}^{z_1,\dots,z_c}$ in Eq. (5.9) is calculated. The step by step result of the procedure is presented in Table 5.5. Also, the particular steps regarding discarding indirect/spurious edges are pictorially illustrated in Fig. 5.5. The significant level of 10 percent is considered for accepting a direct causal connection and if the $SDNDTE_{x_s \to x_t}^{z_1,\dots,z_c} < 0.1$, the corresponding edge is considered indirect or spurious. As can be seen in Fig. 5.5, after applying the procedure proposed in *Algorithm* 6,

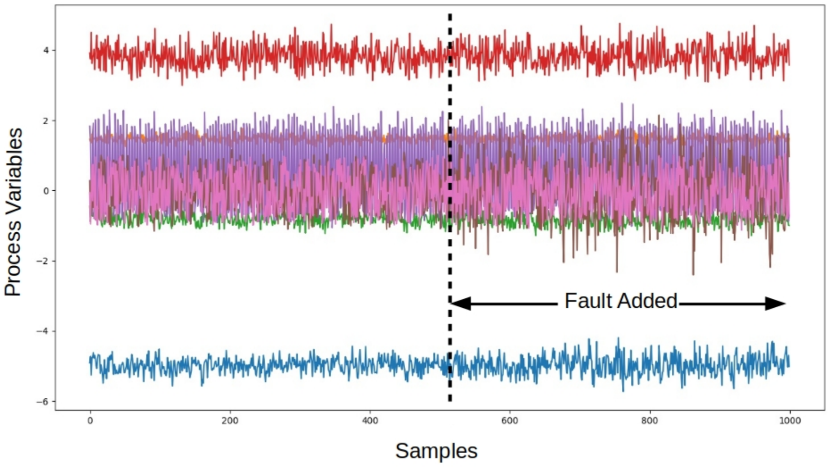


Figure 5.4: Seven process variables of synthetic numerical example between 3500th to 4500th samples.

the pruned causal map between process variables is derived. The next and final step for root-cause fault diagnosis is to apply *Algorithm* 5 and identify the variable(s) causing the anomaly and propagating it through all other variables. Fig. 5.6 shows the detected root-cause fault variables using *Algorithm* 6 and compare it with the true causal map among process variables according to Eq. (5.10). As a result, *Algorithm* 6 could successfully diagnose the root-cause fault variables that is consistent with the variables relationships in Eq. (5.10). However, one would consider the bidirectional edge between x_5 and x_6 as miss-identified direct causal pathway. The reason behind that is the direct causal pathway is a relative concept according to all measured variables and since the true fault F in Eq. (5.10) is not measured, the connection between x_5 and x_6 is considered direct according to available information in measured time series.

The proposed autonomous approach is a general framework in which various causality analysis techniques can be fitted into. Therefore, in order to indicate the improvement in the calculation complexity of the proposed SDNDTE, the conventional way of determining DTE through kernel PDF fitting ([59]) is also utilized for the pruning procedure. Table 5.3 shows the hyper-parameters required to generate embedding vectors for fitting kernel joint-PDF function $\hat{f}(x) = \frac{(det\mathbf{S})^{-0.5}}{N\Gamma q}\sum_{i=1}^{N} K\{\Gamma^{-2}(x - X_i)^T S^{-1}(x - X_i)\}$, where $\Gamma = 1.06N^{-1/(4+q)}$ and S is the covariance matrix of time-series data and K is the *Gaussian* kernel function. Table 5.5 indicates the result of normalized direct transfer entropy (NDTE) which is determined from Eq. (5.9) except in the SDHs are replaced with conventional entropy H. The result of both approaches are consistent except for the $NDTE^{x_3}_{x_6\to x_4} = 0.109$ in which the normalized direct causality is greater than threshold of 0.1 by 0.009 and it contradicts with the SDNDTE results. This inconsistency might be due to the difference in the length of the time series considered

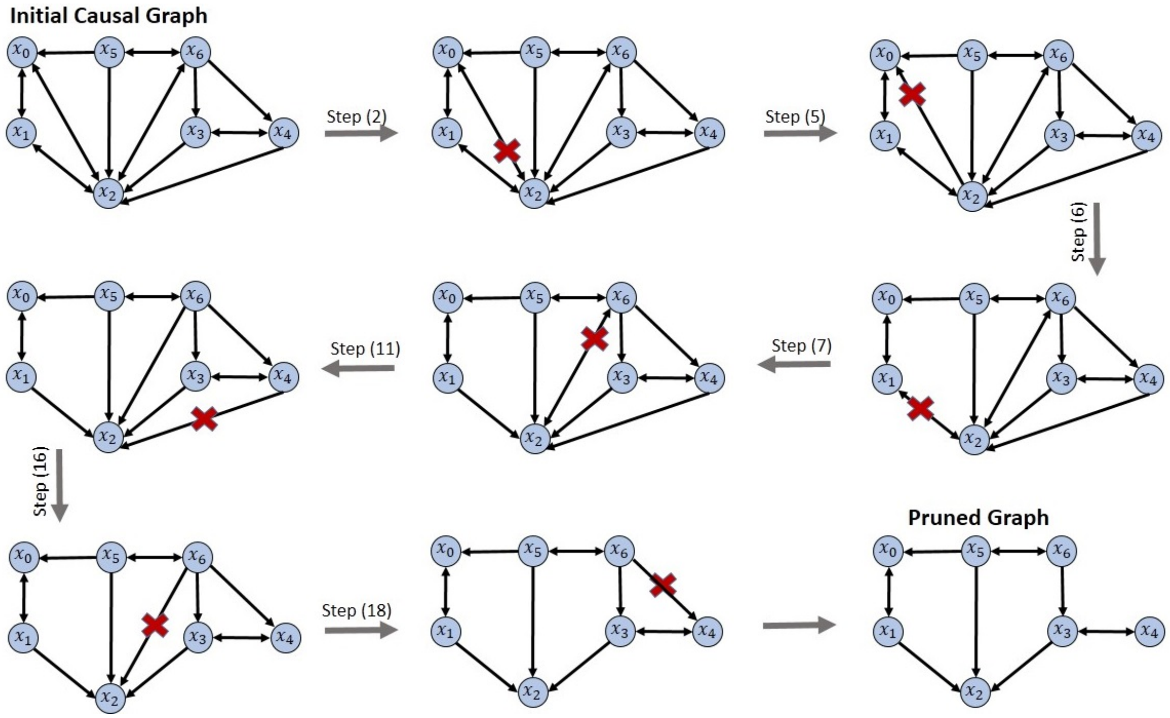


Figure 5.5: The pictorial step by step pruning procedure explained in Table 5.5, which leads to discarding indirect or spurious edges.

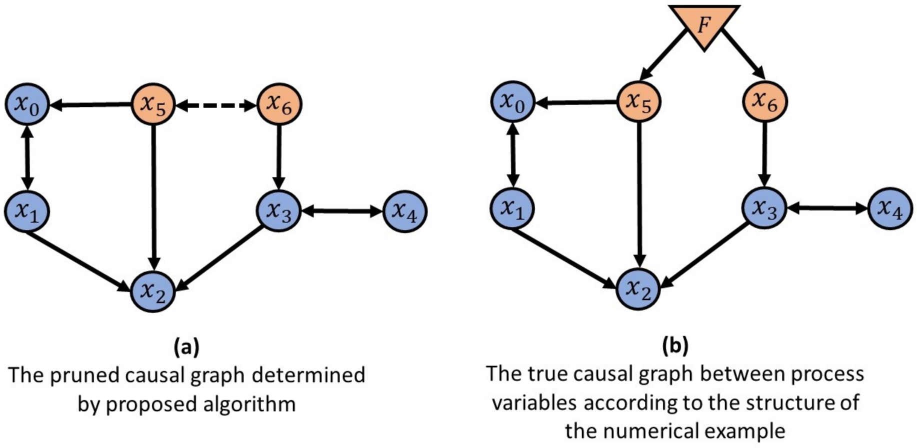

F
x_0
x_1
x_2
x_3
x_4
x_5
x_6
(a)
The pruned causal graph determined by proposed algorithm
(b)
The true causal graph between process variables according to the structure of the numerical example

Figure 5.6: The comparison between the pruned graph which is determined by applying proposed autonomous algorithms, SDNTE and SDNDTE (a) and the actual causal graph found according to the Eq. (5.10).

Table 5.4: The result of calculating $SDNTE_{x_s \to x_t}$ (i.e. x_s row variable to x_t column variable), for all seven variables in numerical example.

Index	x_0	x_1	x_2	x_3	x_4	x_5	x_6
x_0	N.A.	**0.486**	**0.291**	0.025	0.051	0.079	0.081
x_1	**0.386**	N.A.	**0.205**	0.023	0.007	0.082	0.049
x_2	**0.335**	**0.372**	N.A.	0.018	0.023	0.062	**0.211**
x_3	0.001	0.045	**0.298**	N.A.	**0.413**	0.002	0.009
x_4	0.000	0.002	**0.312**	**0.160**	N.A.	0.062	0.015
x_5	**0.431**	0.061	**0.227**	0.024	0.031	N.A.	**0.422**
x_6	0.010	0.035	**0.282**	**0.405**	**0.319**	**0.472**	N.A.

in the calculation of NDTE. It should be noted that this difference will not affect the result of the root-cause fault diagnosis after applying the steps in *Algorithm* 6.

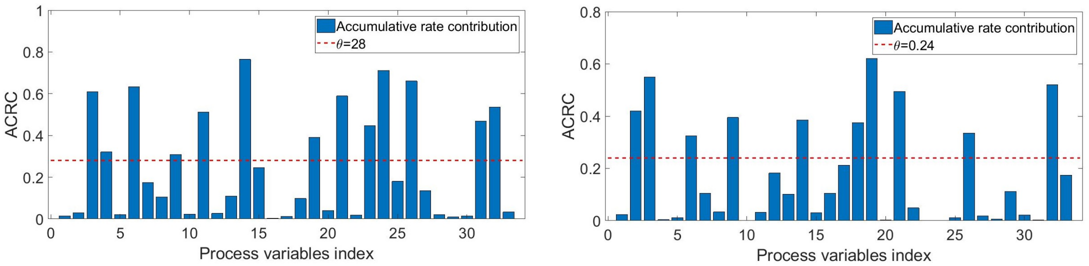
The calculation elapsed time for determining $NDTE$ and $SDNDTE$ are presented in Table 5.5 in the same computational condition (CPU $i7$ @3.2 GHZ and RAM 8 GB). Elapsed time indicates the time that is needed to verify whether the corresponding edge is direct. This case study indicates that the total elapsed time for pruning the initial causal graph using the proposed SDNDTE method is 23.859 second and for conventional DTE method [59] is 354.41 seconds that is showing over 15 times improvement.

Table 5.3: The hyper-parameters required for creating embedding vector prior to fitting joint-pdf function for all seven variables in the numerical example.

variable index	x_0	x_1	x_2	x_3	x_4	x_5	x_6
τ	1	1	1	1	1	1	1
h	1	1	1	1	2	2	2
l	1	2	1	1	2	2	1

Table 5.5: The result of applying *Algorithm* 6 to create a pruned causal graph indicating direct information pathways amongst variables of the numerical example.

Step	$x_s \rightarrow x_t$	$SIVs$	$IIVs$	$SDNDTE^{z_1,\dots,z_c}_{x_s\rightarrow x_t}$	*Elpased seconds using SDNDTE*	$NDTE^{z_1,\dots,z_c}_{x_s\rightarrow x_t}$ *usingPDF fitting*	*Elpased seconds using NDTE*	*Direct Connection checkmark*
(1)	$x_0 \rightarrow x_1$	N.A	x_2	0.432	1.098	0.302	12.34	✓
(2)	$x_0 \rightarrow x_2$	x_5, x_6	x_1	0.029	2.151	0.012	42.65	✗
(3)	$x_1 \rightarrow x_0$	N.A	x_2, x_5	0.510	1.086	0.482	22.03	✓
(4)	$x_1 \rightarrow x_2$	x_5, x_6	N.A	0.305	1.202	0.203	23.89	✓
(5)	$x_2 \rightarrow x_0$	N.A	x_5, x_1	0.036	1.122	0.091	18.66	✗
(6)	$x_2 \rightarrow x_1$	N.A	x_0	0.010	0.964	0.002	13.47	✗
(7)	$x_2 \rightarrow x_6$	x_5	N.A	0.079	1.138	0.006	10.57	✗
(8)	$x_3 \rightarrow x_4$	x_6	N.A	0.615	1.098	0.535	12.18	✓
(9)	$x_3 \rightarrow x_2$	x_5, x_6	x_4	0.436	3.158	0.315	48.56	✓
(10)	$x_4 \rightarrow x_3$	x_6	N.A	0.215	0.814	0.129	11.10	✓
(11)	$x_4 \rightarrow x_2$	x_5, x_6	x_3	0.009	2.476	0.016	40.05	✗
(12)	$x_5 \rightarrow x_0$	N.A	N.A	N.A	N.A	N.A	N.A	✓
(13)	$x_5 \rightarrow x_2$	N.A	x_6, x_3, x_1	0.420	3.204	0.585	36.20	✓
(14)	$x_5 \rightarrow x_6$	N.A	N.A	N.A	N.A	N.A	N.A	✓
(15)	$x_6 \rightarrow x_5$	N.A	N.A	N.A	N.A	N.A	N.A	✓
(16)	$x_6 \rightarrow x_2$	N.A	x_3, x_5, x_1	0.055	2.368	0.041	39.35	✗
(17)	$x_6 \rightarrow x_3$	N.A	x_4	0.215	0.885	0.232	8.91	✓
(18)	$x_6 \rightarrow x_4$	N.A	x_3	0.012	1.095	0.109	14.45	✗
total elpased time					23.859		354.41	



(a) ACRC score of TEP varibles to residual ψ for scenario IDV(8)

(b) ACRC score of TEP varibles to residual ψ for scenario IDV(10)

Figure 5.7: Accumulative rate contribution score proposed in [2] is utilized to determine the relative contribution of each process variables to the combined index.

5.7.2 Tennessee Eastman Process

In this section, the performance and real-time efficiency of the entire proposed root-cause fault diagnosis scheme are under study. To this aim, the Tennessee Eastman process (TEP) [121] is considered, which is a well-known benchmark in the field of process monitoring and fault diagnosis. readers can find the preliminary information about the TEP including the list of variables and simulated fault scenarios in section 4.5

In [102], 15 different known malfunction scenarios (IDV(1-15)) are defined for TEP. But since the SDNDTE approach is proposed to handle (quasi-)stationary time series, only those types of fault scenarios are considered here. In this section, two fault scenarios IDV(8) and IDV(10) are under study to show the performance of the proposed framework. The details of the simulation and fault detection step for these two fault scenarios can be found in section 4.5.

Fig. 4.6(a) shows the combined index for fault scenario IDV(8) that is introducing a random variation in A, B and C streams at 2000^{th} sample. After detecting a fault, accumulative rate contribution (ACRC) [2] is determined for each process variables to diagnose those variables that have a significant relative contribution rate to the fault detection index. Fig. 5.7(a) presents the result of the ACRC analysis for IDV(8), which leads to choosing 13 root-cause fault candidates out of 33 process variables.

After finding the potential root-cause fault candidates, according to the flowchart shown in Fig. 5.1, the next step is to generate an initial causal graph using the proposed SDNTE approach. The parameters are chosen for normal operating condition IDV(0) and summarized in *Appendix* Table 3. As presented in *Algorithm* 6, $SDNDTE_{x_s \to x_t}^{z_1 \cdots z_c}$ proposed in Eq. (5.2) is calculated for each pair of 13 process variables for IDV(8) and if the value of each validation is greater than the significance level of 10%, that pathway is considered as a direct causal edge for the initial causal graph shown in Fig. 5.8. In order to identify the indirect or spurious edges in the initial causal graph, $SDNDTE_{x_S \to x_t}^{z_1 \cdots z_c}$ is determined according to the proposed steps in *Algorithm* 6 and the results are brought in Tables 5.6. The first column in Table 5.6 shows the connection under examination and the second and third columns indicate the IIVs and SIVs,

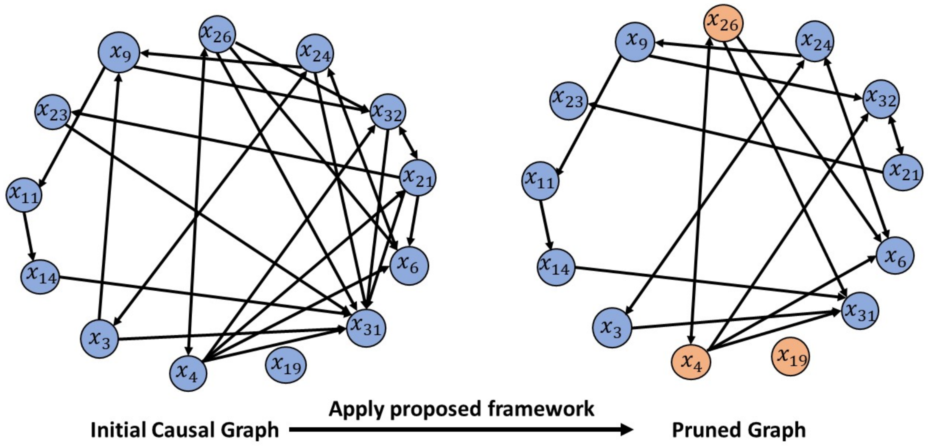


Figure 5.8: The left indicates the result of $SDNTE_{x_s \to x_t}$ with significance level of 10% for TE process and scenario IDV(8). The right shows the pruned causal map after utilizing *Algorithm* 6. The identified potential root-causes are found using *Algorithm* 5 and they are distinguished by different colors.

respectively. The right graph in Fig. 5.8 indicates the pruned causal graph after discarding the edges which do not satisfy the significance level of 10%. By inspection, the proposed SDNDTE approach successfully discards the unnecessary edges in the causal map between process variables to avoid inaccuracy and complication for finding the root-cause variable(s). At this step, *Algorithm* 5 is adopted to autonomously determines the root-cause variables. For fault scenario IDV(8), three potential root-cause candidates x_4 (total feed stream(4)), x_{26} (total feed flow stream(4)) and x_{19} (striper streamflow) are identified in Fig. 5.8. According to the TEP flowchart shown in Fig. 2, if A/B/C composition is subjected to a malfunction which is the case in IDV(8), the fault will directly affect the measurements corresponding to the stream 4. The controller (x_{26}) is acting to compensate the fault and accordingly change the total feed stream x_4 to adjust the set-point, thus, the x_4 and x_{26} are autonomously and successfully diagnosed as the fault root-causes. Also, the stripper stream is directly connected to the stream 4 and variable x_{19} is apparently affected by this fault scenario and it is happened to be an orphan node in the causal graph G.

To further evaluate the effectiveness of the proposed strategy, fault scenario IDV(10) is also considered as a case study. Fig. 4.6(c) shows the combined residual index for this scenario. After detecting a fault, as shown in Fig. 5.7(b), ACRC is utilized to choose 10 root-cause candidates amongst total 33 variables. Then, SDNTE in Eq. (5.2) is applied to find pathways between candidates and generate the initial causal graph shown in Fig. 5.9. Although this initial graph provides intuition regarding the interconnection of the process variables, the spurious and indirect connections might mislead the root-cause diagnosis procedure. Hence, the proposed SDNDTE approach is utilized to test whether each edge in the initial graph

Table 5.6: The result of applying *Algorithm* 6 to create a pruned causal graph which indicates direct causal pathways amongst variables of TEP for scenario IDV(8).

Step	$x_s \rightarrow x_t$	*Source intermediate variables*	*Immediate Intermediate variables*	$SDNDTE^{z_1,\dots,z_c}_{x_s \rightarrow x_t}$	*Direct Connection*
(1)	$x_0 \rightarrow x_7$	$x_4, x_5, x_2, x_{10}, x_3, x_{11}$	N.A	0.001	✗
(2)	$x_1 \rightarrow x_4$	x_2, x_5, x_{10}	N.A	0.212	✓
(3)	$x_1 \rightarrow x_{12}$	N.A	N.A	N.A	✓
(4)	$x_2 \rightarrow x_4$	N.A	x_1, x_5, x_{10}	0.020	✗
(5)	$x_2 \rightarrow x_6$	N.A	x_{10}, x_5	0.254	✓
(6)	$x_2 \rightarrow x_{10}$	N.A	N.A	N.A	✓
(7)	$x_2 \rightarrow x_7$	N.A	$x_3, x_4, x_5, x_9, x_{10}, x_{11}$	**0.219**	✓
(8)	$x_3 \rightarrow x_1$	N.A	x_{11}	0.330	✓
(9)	$x_3 \rightarrow x_7$	x_2, x_{10}	x_4, x_5, x_9, x_{11}	0.004	✗
(10)	$x_3 \rightarrow x_6$	N.A	x_5	0.415	✓
(11)	$x_3 \rightarrow x_{11}$	N.A	N.A	N.A	✓
(12)	$x_4 \rightarrow x_5$	x_2, x_{10}	N.A	0.202	✓
(13)	$x_4 \rightarrow x_7$	x_2, x_{10}	x_5, x_9, x_{11}	0.041	✗
(14)	$x_5 \rightarrow x_6$	x_2, x_3, x_{10}, x_{11}	N.A	0.029	✗
(15)	$x_5 \rightarrow x_7$	$x_1, x_2, x_3, x_{10}, x_{11}$	N.A	0.001	✗
(16)	$x_5 \rightarrow x_0$	N.A	N.A	N.A	✓
(17)	$x_5 \rightarrow x_4$	x_10	N.A	0.515	✓
(18)	$x_6 \rightarrow x_3$	N.A	N.A	N.A	✓
(19)	$x_9 \rightarrow x_7$	x_2, x_3, x_{10}, x_{11}	N.A	0.391	✓
(20)	$x_{10} \rightarrow x_2$	N.A	N.A	N.A	✓
(21)	$x_{10} \rightarrow x_4$	N.A	x_5, x_1	0.501	✓
(22)	$x_{10} \rightarrow x_6$	N.A	x_2	0.515	✓
(23)	$x_{10} \rightarrow x_7$	N.A	x_2, x_9, x_{11}	0.208	✓
(24)	$x_{10} \rightarrow x_5$	N.A	x_4	0.004	✗
(25)	$x_{11} \rightarrow x_7$	x_2, x_{10}	x_9	0.211	✓
(26)	$x_{11} \rightarrow x_1$	N.A	x_3	0.015	✗
(27)	$x_{11} \rightarrow x_3$	N.A	N.A	N.A	✓
(28)	$x_{12} \rightarrow x_9$	N.A	N.A	N.A	✓

represents a direct pathway. Table 5.7 presents the step by step results of SDNDTE analysis of initial edges. According to the results of Table 5.7, the spurious and indirect edges are discarded and the pruned causal graph is generated and shown in Fig. 5.9. Then *Algorithm* 5 is adopted to find the root-cause variables x_{18} (striper temperature) and x_{14} (separator underflow stream 10). By following the interconnection of streams in Fig. 2, fault scenario IDV(10) introduces variation into C feed temperature and directly affects the gas mixture temperature in stream 4 that is monitored by variable x_{18}. Also, due to the closed-loop controller for temperature control of the striper, separator underflow x_{14} is also affected by the source of the fault. At this point, these two variables are the output of the proposed strategy to the domain experts in order to decide the proper maintenance actions.

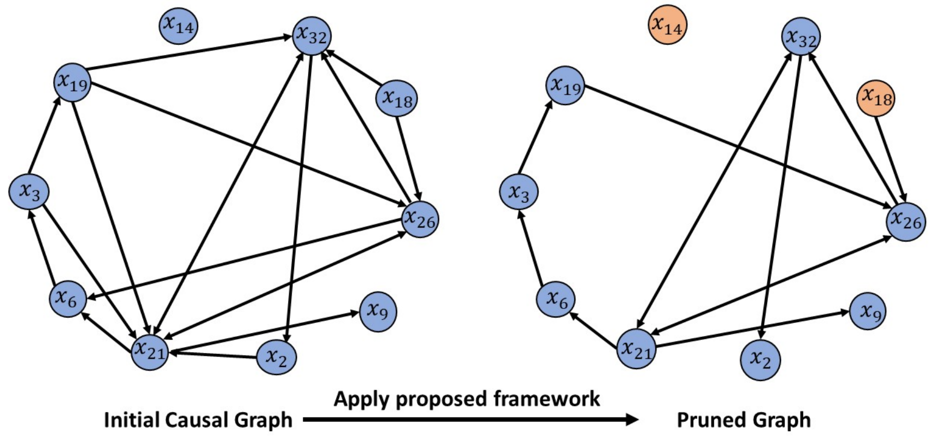


Figure 5.9: The left indicates the result of $SDNTE_{x_s \to x_t}$ with significance level of 10% for TE process and scenario IDV(10). The right shows the pruned causal map after utilizing *Algorithm* 6. The identified root-cause variables found by using *Algorithm* 5 are shown by orange color.

5.8 Summary

According to the existing challenges for real-time root-cause fault diagnosis, an autonomous framework is proposed. The proposed framework is developed to diagnose the root-cause fault of a (non-)linear industrial process which enables quick actions for quality maintenance and safety purpose. To this aim, first, a fault detection approach (e.g. PCA, KPCA, etc.) is conducted to capture the malfunction and a variable screening technique (e.g. RBC, ACRC, etc.) is utilized to choose the potential root-cause candidates. Second, symbolic dynamic normalized transfer entropy (SDNTE) is defined to generate an initial causal map (G) among the candidates. In the third step, the symbolic dynamic filtering approach presented in Chapter 4 is extended to estimate DTE and SDNDTE is proposed to prune indirect/spurious causal edges of the initial map G and generate a pruned causal graph $\bar{G}$ with direct information pathways. Immediate intermediate variable (IIV) and source intermediate variable (SIV) are defined and autonomous efficient algorithms are introduced to identify them without any *a priori* knowledge of process and intervention of an expert. In the last step, a depth first search (DFS)-based algorithm is introduced to locate the root-cause(s) of the fault in the pruned graph. The proposed SDNDTE has less calculation complexity with the conventional way of determining DTE which enables the real-time application of TE for root-cause fault diagnosis. Moreover, this framework eliminates the requirement for process knowledge and expert inspection. It should be noted that the proposed general strategy in this paper is not tied to any particular FD method and causality technique and the individual components presented in Fig. 5.1 can be replaced with respect to the nature of the process and its malfunctions. Fi-

Table 5.7: The result of applying *Algorithm* 6 to create a pruned causal graph which indicates direct causal pathways amongst variables of TEP for IDV(10).

Step	$x_s \to x_t$	*Source intermediate variables*	*Immediate Intermediate variables*	$SDNDTE_{x_s \to x_t}^{z_1,\ldots,z_c}$	*Direct Connection*
(1)	$x_0 \to x_1$	N.A	x_8	0.015	✗
(2)	$x_1 \to x_2$	N.A	x_8	0.305	✓
(3)	$x_1 \to x_6$	N.A	x_8, x_4	0.445	✓
(4)	$x_1 \to x_8$	x_7	x_4	0.215	✓
(5)	$x_1 \to x_9$	N.A	N.A	N.A	✓
(6)	$x_2 \to x_3$	N.A	N.A	N.A	✓
(7)	$x_3 \to x_1$	N.A	x_4, x_6, x_8	0.002	✗
(8)	$x_3 \to x_4$	N.A	N.A	N.A	✓
(9)	$x_4 \to x_1$	N.A	x_6, x_8	0.035	✗
(10)	$x_4 \to x_6$	N.A	x_1, x_8	0.080	✗
(11)	$x_4 \to x_8$	x_1, x_7	N.A	0.386	✓
(12)	$x_6 \to x_0$	N.A	N.A	N.A	✓
(13)	$x_6 \to x_1$	x_8, x_7	N.A	0.513	✓
(14)	$x_7 \to x_6$	N.A	x_1, x_8	0.077	✗
(15)	$x_7 \to x_8$	N.A	N.A	N.A	✓
(16)	$x_8 \to x_1$	N.A	x_6	0.012	✗
(17)	$x_8 \to x_2$	N.A	x_1	0.045	✗
(18)	$x_8 \to x_6$	x_1	N.A	0.194	✓

nally, successful applications of the proposed strategy on a numerical simulation and Tennessee Eastman Process (TEP) are presented.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Industrial processes are often subjected to anomalies that need to be detected and require treatment. These abnormal conditions deteriorate the quality of the process and increase operational costs and possibly lead to hazardous consequences. To this aim, this thesis proposes solutions for some of the ongoing challenges in industrial process health monitoring.

Chapters 2 and 3 are dedicated to remedying the fault detection in non-stationary processes, in which the measurement time-series follow a probability distribution with time-varying mean and constant variances. Although there exist numerous approaches that are based on projection to latent variables such as principal component analysis (PCA) and partial least-squares (PLS), they require the time-series to be stationary with a constant mean and variance. On the other hand, PCA and PLS are well-recognized by industrial operators due to their easy implementation and powerful basis for handling the high-dimensional processes. Although there are adaptive/recursive solutions that update the base-line model in a real time manner, they suffer from on-line computational complexity. Hence, this became a motivation to propose a moving-mean PCA (MM-PCA) approach in Chapter 2, which is not limited to only stationery cases and it does not include any heavy online adaptation. This approach considers the upper bounds of expected range of variations for the process measurements and updates the mean values of the measurements which are utilized in normalization of a new test data. Accordingly, three feature indices are introduced using MM-PCA which monitors the behavior of the time-series variations. The first feature Φ^0_{MM} is the zero-order error index in the definition of ordinary PCA [12]. Furthermore, Φ^1_{MM} and Φ^2_{MM} are the first- and second-order error indices which were defined to monitor the trend pattern of the first and second-order difference of the time-series. These three feature indices are defined and used to propose an overall health index using the concept of kernel density estimator (KDE), which helps process operators to distinguish normal (i.e. no-fault exist) non-stationary mean variation cases form faulty operating conditions. Even though the proposed overall health index may be utilized directly for process health monitoring, an alarm-based algorithm is further suggested which provides caution and fault alarms to assist operators for adopting proper preventive actions.

One of the fundamental differences between the adaptive PCA and the proposed MM-PCA approach is how, i.e. based on what criteria, to recalibrate the mean value of the test data when the process is subjected to a new variation. In MM-PCA, proper distance-based criteria along with the upper bound of expected variations for time-series are utilized to remove the need for real-time recalibration of the base-line model. Considering knowledge about the upper bound of all process time-series is difficult to achieve in some cases, we propose an analytical solution for finding the unknown upper bounds. Another assumption in the proposed MM-PCA is that the loading vectors for the process time-series remain the same during the normal non-stationary mean changes. On the contrary, in adaptive PCA, there is no assumption regarding the consistency of the loading vectors and a real-time recalibration mechanism will update the base-line loading vectors with the price of applying singular value decomposition for each new batch of test data.

Each process may have one or some process measurements that represent the quality of the normal operating condition. These measurements are also considered as key performance indicators (KPIs) and play a prominent role in the field of process control and monitoring. For the latter case, domain experts are often more interested to conduct fault detection and simultaneously monitor the impact of the fault on the process quality outputs. To this aim, PLS-based approaches have been commonly applied for the stationary processes. Moreover, adaptive PLS methods are developed to update the baseline structure when the process time-series are subjected to non-stationary variations. However, this solution requires online parameter adaptation which introduces heavy computational complexity. To this aim, a PLS-based approach is proposed in Chapter 3 without an online update mechanism applicable for a process that is subjected to non-stationary mean changes in their measurements. This method leverages from the orthogonal projection of time-series into quality output-related and unrelated components. Furthermore, the concept of principal manifolds is adopted to model the time-varying relationship between the projected loading directions with respect to the normal changes that should not be miss-detected as a fault. The performance of the proposed formulation is shown using numerical synthetic simulation and continuous stirred tank reactor (CSTR) process.

When calculation complexity is a concern, the proposed PLS-based approach is an alternative for adaptive PLS [47] method. However, there exist some assumptions such as it assumes the regression model between process time-series and the quality output remains unchanged during the non-stationary mean variations. Also, some of the normal non-stationary changes should be included in the training data-set. It should be mentioned that this approach can be considered as an unsupervised technique which do not require any faulty data in the training phase. On the other hand, there exist some methods such as Fisher discriminate analysis (FDA) [122] which create a bank of base-line models for both normal and faulty conditions.

Chapters 2 and 3 give solutions for performing anomaly detection in non-stationary processes, which is often the first main step for conducting a through process monitoring. After detecting the fault in a process, knowledge about the source of it is crucial. Hence, domain experts are motivated to perform root-cause analysis approaches to identify the process measurements that are the source of the fault propagating into other counterpart. This information helps the operators to diagnose the abnormality and therefore identify the faulty process component(s). One of the recent solutions to identify the root-cause of the fault is based on performing causality analysis among the process time-series to find the information pathways. In Chapter 4, it is proposed to utilize transfer entropy (TE) as a viable tool for measuring the information inference between time-series which may have (non-)linear relationships. Although TE has been recently utilized for this purpose, the its conventional estimation approach which is based on kernel density estimators suffers from computational complexity and can not be used for real-time causality analysis purposes. To this aim, symbolic dynamic filtering (SDF) is utilized to define symbolic dynamic transfer entropy (SDTE) that has less computational complexity in comparison with the conventional KDE approach. The SDF concept is applied to define the joint *XD-Markov* machines to estimate the joint *Shannon* entropies in the definition of TE. Moreover, the general framework proposed in Chapter 4 requires less number of TE estimation to identify the root-cause fault variable(s) among the potential candidates. The efficiency and contribution of the proposed approach are presented by applying it to the Tennessee Eastman Process (TEP) benchmark. Furthermore, this method is applied to an industrial centrifuge that suffers from nozzle plugging issue and its operators were interested to identify the root-cause of the fault to apply proactive maintenance actions.

Although TE can identify the presence a causal inference between two time-series, it can not guarantee that the causality is direct or due to one/some intermediate variables. To this aim, direct TE (DTE) has been used to reveal the spurious and indirect causal pathways among the process variables. In Chapter 5, application of SDF is extended to define multi-dimensional joint *XD-Markov* to propose symbolic dynamic direct transfer entropy $SDDTE_{x_a \rightarrow x_b}^{z_1 \cdots z_c}$ from x_a to x_b with presence of intermediate variables $z_1, ..., z_c$. This contribution let the operators replace the conventional KDE approach for estimating DTE and incorporate it for real-time causality analysis purposes. The proposed SDNTE in Chapter 4 and SDNDTE in Chapter 5 enables the application of TE and DTE for real-time root-cause fault diagnosis for early treatment of the detected abnormality in a process.

Autonomous algorithms can be applied to reduce the complex parameter tuning and reduce (or eliminate) the need for *a-priori* knowledge and domain expert intervention. Also, the structure of using the available monitoring tools such as SDNTE and SDNDTE can be properly adopted in such a way to reduce the manual selection/intervention of an operator during the process monitoring. In Chapter 5, a general schema is proposed to autonomously identify the root-cause fault variable(s). This framework assumes that the proper fault detection approach is in place to detect the fault upon existence. Then, a complementary fault diagnosis approach is adopted to select the process measurements affected by the fault. In the next step, SDNTE is used to create an initial directed graph G that represents the connection among the process candidate variables. Then, SDNDTE is used to validate if each edge in graph G is indirect or spurious. For this purpose, it is required to efficiently identify which surrounding variable may possibly act as an intermediate counterpart to infer the indirect/spurious edge. To this aim, first the concept of immediate intermediate variables (IIV) and source intermediate variables (SIV) are defined and then topological algorithms are developed to efficiently select them for each edge in the graph G. This part of the proposed framework eliminates the intervention of the process operators or any required process knowledge. After pruning the edges that do not represent a direct causal interaction, a depth first search (DFS)-based algorithm is developed to select the root-cause fault variable(s). The output of the proposed framework lets the operators realized which measurements are the cause of the detected fault in the process. The performance of the proposed approach in Chapter 5 is tested on a synthetic numerical example and TEP. Also, the computational efficiency of the proposed SDNTE and SDNDTE approach in comparison with the conventional KDE-based approach is compared.

6.2 Future Work

This thesis provides solutions to some of the limitations in two main steps of process health monitoring; fault detection and autonomous root-cause fault diagnosis. However, there exist more potential research that can be conducted to increase the accuracy and implementation of the process monitoring strategies for different case scenarios. As the future work plan, machine learning (ML) and artificial intelligence (AI) are the main tools to achieve the objectives.

Data-driven modeling attracts great attention and has been widely applied due to the fast development of ML and big-data analytics. It is known that ML encompasses the more traditional multivariate statistical analysis methods, such as principal component analysis (PCA), partial least-squares (PLS) and their variants. AI-based deep learning (DL) has gained significant interests in the field of industrial process health monitoring such as fault detection and diagnosis, owing to its inherent ability to handle uncertainties and non-linear transformation for data with any distributions (e.g. non-Gaussian).

As the future work of this thesis, we investigate various machine learning techniques to specifically handle non-linear transformation and classification, and when the data exhibits non-stationary trends. Methods including random forest [123], autoencoder (AE) [124], deep belief network (DBN) [125], deep boltzman machines (DBM) [126], convolutional neural network (CNN) [127] and recurrent neural network(RNN) have been applied and implemented to detect and diagnose anomalies in different industrial processes and systems.

Two main objectives are set forward as the future work of the research proposed in this thesis.

- **Objective 1: Application of Recommender System for Fault Diagnosis**

The conventional way of conducting fault detection and diagnosis for process monitoring purposes is to firstly select and apply a method on normal operating modes and different fault scenarios. Then by utilizing a particular criterion (e.g. false alarm rate, RMS value, etc.), the accuracy of the method is measured and it is evaluated by the domain expert or cross-validation approach. This procedure that is so-called pipe-line testing is repeated for different methodologies and the best performance is selected as a proper method for the problem under study. One of the limitations of this schema is that sometimes the selected method may not be the right fit for the next upcoming test data or test scenario because it was not meet the statistical properties of that specific case scenario. Moreover, each method has its own pros and cons, hence, for different operating/fault scenarios of a process, a single method may not have the best performance. This motivates us to propose a framework to aggregate different approaches for the same fault detection/diagnosis task on a process.

This can be done by application of the content-based recommender systems (i.e. categorical AE). The idea for this objective is that the concept of recommender system can be utilized in a way that each one of p different approaches (i.e. assume that there are p methods for conducting a fault detection/diagnosis task) is deemed as a *voter* and each fault scenario (i.e. assume that N different operating mode, fault scenario exist, process mode, etc.) is an *object*. Then each user will have a normalized *vote* for an *object* that is the result of applying that particular method on the corresponding data-set. Then the goal is to create m number of fictitious categorical features from an adequate training data-set and create an interconnected relationship between the votes and those categorical features. This formulation similarly used in the NETFLIX platform as a movie recommender module and the system is successfully working for that purpose and can be considered as a proof of concept as a potential solution to the proposed objective.

• Objective 2: Hybrid Twin-Model for Fault Detection in Non-Stationary Processes

One of the ongoing challenges in process monitoring is the accuracy of the data model, also referred to as 'digital-twin' (of the actual process/plant). In addition, prognosis based on KPI prediction is of great interests as discussed in Chapter 3. Plant data collected in real-time generally is non-stationary that may represent different operating conditions. As a result, a model built upon such data may suffer from inaccuracy and fidelity, since it is considered to be an average model for multiple operating conditions. In addition, noises and disturbances are also sources of modeling errors. In existing works, data models are built by applying multivariate statistical analysis approaches, which usually impose assumptions of linearity, stationarity, and *Gaussian* distribution. Efforts have been made in this thesis to tackle practical applications (when these assumptions are violated), by adding mechanisms to detect the nominal change of statistical characteristics and modify baseline model parameters accordingly, leading to an unsupervised learning based fault detection approach [128] [129]. Along with this objective, we will also investigate various supervised learning approaches and their applications to fault diagnosis and prognosis.

There exist numerous software systems that include sophisticated and scalable simulation systems to emulate various process components, equipment, and units, based on first-principle models and laboratory experiments. These systems can be referred to as the operator's training system (OTS), which can be used to simulate and test numerous operations including both

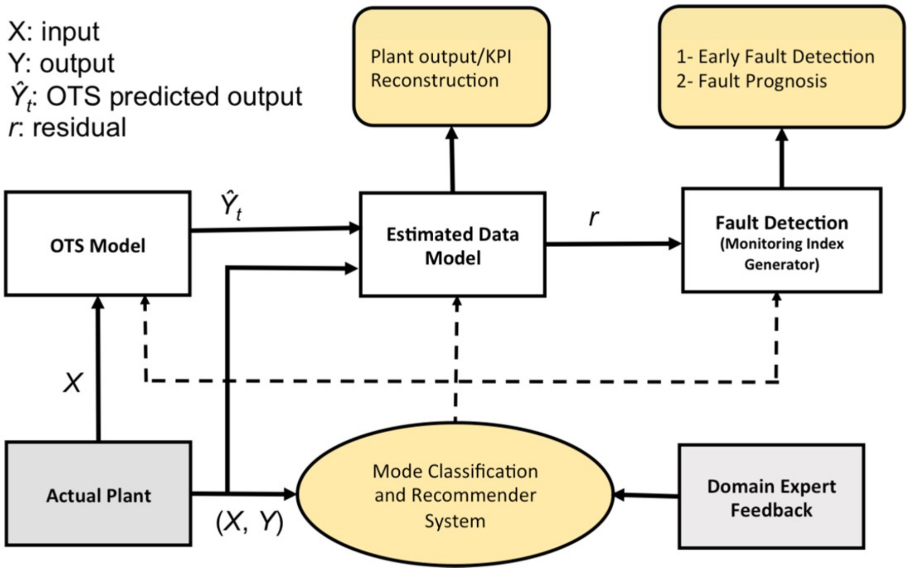


Figure 6.1: The future workflow of the proposed objective for the digital-twin platform

normal and faulty ones. These systems are under constant tuning, testing, and customization, and are considered to be of high fidelity. Hence an initiative has been proposed to utilize the data produced by an OTS for the training of the digital-twin (data) model. More specifically, plant data and OTS data will be integrated into supervised learning for early fault detection and prognosis.

A schematic diagram is given in Fig. 6.1 to show the proposed integrated digital-twin model, and the fault diagnosis and prognosis modules. The advantage of incorporating the OTS model software is twofold. First, it can provide more valuable training data by simulating different operation modes, including normal ones, fluctuations in manipulated variables, and even some failure modes in equipment. Such data may not always be available from the real plant but are essential for the successful application of data-driven and machine learning techniques. Secondly, OTS works similarly as a first-principle model, providing redundancy in the fault diagnosis and health monitoring system. It can also be used to tune the baseline model and threshold for the fault detector. Notwithstanding the obvious advantage, OTS cannot completely replace the data model due to the following challenges:

1) OTS is an offline process and cannot produce online data in real-time;

2) Certain physical scenarios are not modeled in OTS, such as vibration dynamics of rotating equipment, and slow changes of physical parameters due to decaying, wearing and corrosions.

OTS data and plant data should be fused and utilized for diagnosis and prognosis. We use the measured actual plant input X and feed it into the OTS to reconstruct output $\hat{Y}_t$ periodically (offline), which can be used together with actual measurements to train localized base-line models. To achieve this, regression techniques or autoencoder can be applied, which are suitable for online output reconstruction under different modes of operation. A residual signal can then be generated for fault detection and analytics in the next step. In order to tackle multiple operations of the processes, a mode classification and recommender system need to be designed, which operates at both pre-processing and post-processing stages and is responsible for parameter updating and adaptation.

Bibliography

[1] J. Jiao, H. Yu, and G. Wang, “A quality-related fault detection approach based on dynamic least squares for process monitoring,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2625–2632, 2016.

[2] K.-X. Peng, K. Zhang, and G. Li, “Online contribution rate based fault diagnosis for nonlinear industrial processes,” *Acta Automatica Sinica*, vol. 40, no. 3, pp. 423 – 430, 2014.

[3] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, “A survey of fault detection, isolation, and reconfiguration methods,” *IEEE Trans. Control Syst. Tech*, vol. 18, no. 3, pp. 636–653, 2010.

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

[5] V. Venkatasubramanian, R. Rengaswamy, and S. Kavuri, “A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies,” *J. Comp. Chem. Eng.*, vol. 27, no. 3, pp. 313 – 326, 2003.

[6] L. Li, M. Chadli, S. X. Ding, J. Qiu, and Y. Yang, “Diagnostic observer design for t-s fuzzy systems:application to real-time weighted fault detection approach,” *IEEE Trans. Fuzzy Systems*, 2017.

[7] T. Youssef, M. Chadli, H. Karimi, and R. Wang, “Actuator and sensor faults estimation based on proportional integral observer for {TS} fuzzy model,” *Journal of the Franklin Institute*, vol. 354, no. 6, pp. 2524 – 2542, 2017.

[8] A. Chibani, M. Chadli, P. Shi, and N. B. Braiek, “Fuzzy fault detection filter design for t-s fuzzy systems in finite frequency domain,” *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[9] S. Yin, S. Ding, A. Haghani, H. Hao, and P. Zhang, “A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process,” *J. Process Control*, vol. 22, no. 9, pp. 1567 – 1581, 2012.

[10] Y. Wang, G. Ma, S. Ding, and C. Li, “Subspace aided data-driven design of robust fault detection and isolation systems,” *Automatica*, vol. 47, no. 11, pp. 2474 – 2480, 2011.

[11] S. Ding, P. Zhang, A. Naik, E. Ding, and B. Huang, “Subspace method aided data-driven design of fault detection and isolation systems,” *J. Process Control*, vol. 19, no. 9, pp. 1496 – 1510, 2009.

[12] S. Qin, “Survey on data-driven industrial process monitoring and diagnosis,” *Annual Reviews in Control*, vol. 36, no. 2, pp. 220 – 234, 2012.

[13] L. Chian, R. Braatz, and E. Russell, “Fault detection and diagnosis in industrial system,” *New York, NY, USA, Springer*, 2001.

[14] C. Alcala and S. J. Qin, “Reconstruction-based contribution for process monitoring,” *Automatica*, vol. 45, no. 7, pp. 1593 – 1600, 2009.

[15] R. Landman and S.-L. Jämsä-Jounela, “Hybrid approach to casual analysis on a complex industrial system based on transfer entropy in conjunction with process connectivity information,” *Control Engineering Practice*, vol. 53, pp. 14 – 23, 2016.

[16] K. Kolcio and L. Fesq, “Model-based off-nominal state isolation and detection system for autonomous fault management,” in *2016 IEEE Aerospace Conference*, 2016, pp. 1–13.

[17] A. Keipour, M. Mousaei, and S. Scherer, “Automatic real-time anomaly detection for autonomous aerial vehicles,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5679–5685.

[18] G. Li, J. Qin, and T. Yuan, “Nonstationarity and cointegration tests for fault detection of dynamic processes,” *IFAC Proceedings of the 19th World Congress*, pp. 24–29, 2014.

[19] S. W. Choi, E. B. Martin, A. J. Morris, and I.-B. Lee, “Fault detection based on a maximum-likelihood principal component analysis (pca) mixture,” *Industrial & Engineering Chemistry Research*, vol. 44, no. 7, pp. 2316–2327, 2005.

[20] B. Rashidi, D. S. Singh, and Q. Zhao, “Data-driven root-cause fault diagnosis for multivariate non-linear processes,” *Control Engineering Practice*, vol. 70, pp. 134 – 147, 2018.

[21] S. Johansen, “Statistical analysis of cointegration vectors,” *Journal of Economic Dynamics and Control*, vol. 12, pp. 231–254, 1988.

[22] Q. Chen, U. Kruger, and A.-Y.-T. Leung, “Cointegration testing method for monitoring nonstationary processes,” *Ind. Eng. Chem. Res*, vol. 7, no. 48, pp. 3533–3543, 2009.

[23] W. Ku, R. Storer, and C. Georgakis, “Disturbance detection and isolation by dynamic principal component analysis,” *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 179 – 196, 1995.

[24] L. Ma, J. Dong, K. Peng, and K. Zhang, "A novel data-based quality-related fault diagnosis scheme for fault detection and root cause diagnosis with application to hot strip mill process," *Control Engineering Practice*, vol. 67, pp. 43 – 51, 2017.

[25] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[26] S. Yin, S. X. Ding, P. Zhang, A. Hagahni, and A. Naik, "Study on modifications of pls approach for process monitoring," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 389 – 12 394, 2011.

[27] S. Wold, K. Esbensen, and P. Geladi, "Proceedings of the multivariate statistical workshop for geologists and geochemists principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, pp. 37 – 52, 1987.

[28] T. Rato and M. Reis, "Fault detection in the tennessee eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (dpca-dr)," *Chemom. Intell. Lab. Syst.*, vol. 125, no. 1, pp. 101 – 108, 2013.

[29] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[30] M. Ding, Z. Tian, and H. Xu, "Adaptive kernel principal component analysis," *J. Signal Processing*, vol. 90, no. 5, pp. 1542 – 1553, 2010.

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

[32] H. Hoffmann, "Kernel {PCA} for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863 – 874, 2007.

[33] J.-H. Cho, J.-M. Lee, S. W. Choi, D. Lee, and I.-B. Lee, "Fault identification for process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 60, no. 1, pp. 279 – 288, 2005.

[34] N. Kwak, "Nonlinear projection trick in kernel methods: An alternative to the kernel trick," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 2113–2119, 2013.

[35] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica Chimica Acta*, vol. 185, no. Supplement C, pp. 1 – 17, 1986.

[36] Z. Hu, Z. Chen, W. Gui, and B. Jiang, "Adaptive pca based fault diagnosis scheme in imperial smelting process," *ISA Transactions*, vol. 53, no. 5, pp. 1446 – 1455, 2014.

[37] C. Stork, D. Veltkamp, and B. Kowalski, “Identification of multiple sensor disturbances during process monitoring,” *Analytical Chemistry*, vol. 69, no. 24, pp. 5031–5036, 1997.

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

[39] K. Helland, H. E. Berntsen, O. S. Borgen, and H. Martens, “Recursive algorithm for partial least squares regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 14, no. 1, pp. 129 – 137, 1992.

[40] S. Yin, S. X. Ding, P. Zhang, A. Hagahni, and A. Naik, “Study on modifications of pls approach for process monitoring,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 389 – 12 394, 2011, 18th IFAC World Congress.

[41] S. J. Qin and Y. Zheng, “Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures,” *AIChE Journal*, vol. 59, no. 2, pp. 496–504, 2013.

[42] D. Zhou, G. Li, and S. J. Qin, “Total projection to latent structures for process monitoring,” *AIChE Journal*, vol. 56, no. 1, pp. 168–178, 2010.

[43] S. Yin, X. Zhu, and O. Kaynak, “Improved pls focused on key-performance-indicator-related fault diagnosis,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1651–1658, 2015.

[44] Q. Jia and Y. Zhang, “Quality-related fault detection approach based on dynamic kernel partial least squares,” *Chemical Engineering Research and Design*, vol. 106, pp. 242 – 252, 2016.

[45] K. Peng, K. Zhang, B. You, and J. Dong, “Quality-related prediction and monitoring of multi-mode processes using multiple pls with application to an industrial hot strip mill,” *Neurocomputing*, vol. 168, pp. 1094 – 1103, 2015.

[46] A. Haghani, T. Jeinsch, and S. X. Ding, “Quality-related fault detection in industrial multimode dynamic processes,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6446–6453, 2014.

[47] J. Dong, K. Zhang, Y. Huang, G. Li, and K. Peng, “Adaptive total pls based quality-relevant process monitoring with application to the tennessee eastman process,” *Neurocomputing*, vol. 154, pp. 77 – 85, 2015.

[48] Y. Gao, X. Wang, Z. Wang, and L. Zhao, “Fault detection in time-varying chemical process through incremental principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 158, pp. 102 – 116, 2016.

[49] P. Kerkhof, J. Vanlaer, G. Gins, and J. V. Impe, “Contribution plots for statistical process control: Analysis of the smearing-out effect,” in *Control Conference (ECC)*, Jul. 2013, pp. 428–433.

[50] G. Li, S. J. Qin, and T. Yuan, “Data-driven root cause diagnosis of faults in process industries,” *Chemometrics and Intelligent Laboratory Systems*, vol. 159, pp. 1 – 11, 2016.

[51] H. Jiang, M. S. Choudhury, and S. L. Shah, “Detection and diagnosis of plant-wide oscillations from industrial data using the spectral envelope method,” *Journal of Process Control*, vol. 17, no. 2, pp. 143 – 155, 2007.

[52] H. Jiang, R. Patwardhan, and S. L. Shah, “Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix,” *Journal of Process Control*, vol. 19, no. 8, pp. 1347 – 1354, 2009.

[53] G. Weidl, A. Madsen, and S. Israelson, “Applications of object-oriented bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes,” *Computers and Chemical Engineering*, vol. 29, no. 9, pp. 1996 – 2009, 2005.

[54] T. Yuan and S. J. Qin, “Root cause diagnosis of plant-wide oscillations using granger causality,” *Journal of Process Control*, vol. 24, no. 2, pp. 450 – 459, 2014.

[55] G. Li, S. J. Qin, and T. Yuan, “Data-driven root cause diagnosis of faults in process industries,” *Chemometrics and Intelligent Laboratory Systems*, vol. 159, pp. 1 – 11, 2016.

[56] T. Schreibers, “Measuring information transfer,” *Phys. Rev. Lett.*, vol. 85, pp. 461–464, 2000.

[57] M. Bauer, J. Cox, M. Caveness, J. Downs, and N. Thornhill, “Finding the direction of disturbance propagation in a chemical process using transfer entropy,” *IEEE Trans. Control. Syst. Tech*, vol. 15, no. 1, pp. 12–21, 2007.

[58] R. Landman and S.-L. Jämsä-Jounela, “Hybrid approach to casual analysis on a complex industrial system based on transfer entropy in conjunction with process connectivity information,” *Control Engineering Practice*, vol. 53, pp. 14 – 23, 2016.

[59] P. Duan, F. Yang, T. Chen, and S. Shah, “Direct causality detection via the transfer entropy approach,” *IEEE Trans. Control Syst. Tech*, vol. 21, no. 6, pp. 2052–2066, 2013.

[60] P. Duan, F. Yang, S. Shah, and T. Chen, “Transfer zero-entropy and its application for capturing cause and effect relationship between variables,” *IEEE Trans. Control Syst. Tech*, vol. 23, no. 3, pp. 855–867, 2015.

[61] B. Silverman, “Density estimation for statistics and data analysis,” *Chapman & Hall/CRC*, 1986.

[62] L. Ma, J. Dong, K. Peng, and K. Zhang, "A novel data-based quality-related fault diagnosis scheme for fault detection and root cause diagnosis with application to hot strip mill process," *Control Engineering Practice*, vol. 67, pp. 43 – 51, 2017.

[63] R. Landman, J. Kortela, Q. Sun, and S.-L. Jämsä-Jounela, "Fault propagation analysis of oscillations in control loops using data-driven causality and plant connectivity," *Computers and Chemical Engineering*, vol. 71, pp. 446 – 456, 2014.

[64] L. Ma, J. Dong, and K. Peng, "Root cause diagnosis of quality-related faults in industrial multimode processes using robust gaussian mixture model and transfer entropy," *Neurocomputing*, vol. 285, pp. 60 – 73, 2018.

[65] X. Liu, U. Kruger, L. Xie, and S. Wang, "Moving window kernel pca for adaptive monitoring of nonlinear process," *Chemom Intell Lab Syst*, vol. 96, no. 2, pp. 132 – 143, 2009.

[66] I. Jolliffe, "Principal component analysis," *Wiley*, 2014.

[67] H. H. Yue and S. J. Qin, "Reconstruction-based fault identification using a combined index," *Industrial & Engineering Chemistry Research*, vol. 40, no. 20, pp. 4403–4414, 2001.

[68] S. X. Ding, "Model-based fault diagnosis techniques: Design schemes, algorithms and tools," *Springer*, 2013.

[69] B. Silverman, "Density estimation for statistics and data analysis," *New York: Routledge*, 1986.

[70] Y. Cheng, "Data-driven techniques on alarm system analysis and improvement," *Ph.D. Thesis, University of Alberta, ECE*, 2013.

[71] S. Lai, "Data-driven methods for industrial alarm flood analysis," *Ph.D. Thesis, University of Alberta, ECE*, 2017.

[72] J. A. Westerhuis, T. Kourti, and J. F. MacGregor, "Analysis of multiblock and hierarchical pca and pls models," *Journal of Chemometrics*, vol. 12, no. 5, pp. 301–321, 1998.

[73] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[74] S. Yoon and J. F. MacGregor, "Fault diagnosis with multivariate statistical models part i: using steady state fault signatures," *Journal of Process Control*, vol. 11, no. 4, pp. 387 – 400, 2001.

[75] M. S. Reis and G. Gins, "Industrial process monitoring in the big data/industry 4.0 era: from detection, to diagnosis, to prognosis," *Processes*, vol. 5, no. 3, 2017.

[76] P. Hajihosseini, K. Salahshoor, and B. Moshiri, “Process fault isolation based on transfer entropy algorithm,” *ISA Transactions*, vol. 53, no. 2, pp. 230 – 240, 2014.

[77] T. Yuan and S. J. Qin, “Root cause diagnosis of plant-wide oscillations using granger causality,” *Journal of Process Control*, vol. 24, no. 2, pp. 450 – 459, 2014.

[78] M. Bauer and N. F. Thornhill, “A practical method for identifying the propagation path of plant-wide disturbances,” *Journal of Process Control*, vol. 18, no. 7, pp. 707 – 719, 2008.

[79] L. Zhang, J. Zheng, and C. Xia, “Propagation analysis of plant-wide oscillations using partial directed coherence,” *Journal Of Chemical Engineering Of Japan*, vol. 48, no. 9, pp. 766–773, 2015.

[80] L. Luo, F. Cheng, T. Qiu, and J. Zhao, “Refined convergent cross-mapping for disturbance propagation analysis of chemical processes,” *Computers and Chemical Engineering*, vol. 106, pp. 1 – 16, 2017.

[81] B. Lindner, L. Auret, M. Bauer, and J. Groenewald, “Comparative analysis of granger causality and transfer entropy to present a decision flow for the application of oscillation diagnosis,” *Journal of Process Control*, vol. 79, pp. 72 – 84, 2019.

[82] C. S. Daw, C. E. A. Finney, , and E. R. Tracy, “A review of symbolic analysis of experimental data,” *Review of Scientific Instruments*, vol. 74, no. 2, pp. 915–930, 2003.

[83] D. Lind, B. Marcus, L. Douglas, and M. Brian, “An introduction to symbolic dynamics and coding,” *Cambridge University Press*, vol. Xi, 1995.

[84] A. Ray, “Symbolic dynamic analysis of complex systems for anomaly detection,” *J. Signal Processing*, vol. 84, no. 7, pp. 1115 – 1130, 2004.

[85] S. C. Chin, A. Ray, and V. Rajagopalan, “Symbolic time series analysis for anomaly detection: A comparative evaluation,” *Signal Processing*, vol. 85, no. 9, pp. 1859 – 1868, 2005.

[86] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco, “Probabilistic finite-state machines - part i,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1013–1025, 2005.

[87] V. Rajagopalan and A. Ray, “Symbolic time series analysis via wavelet-based partitioning,” *J. Signal Processing*, vol. 86, no. 11, pp. 3309–3320, 2006.

[88] D. S. Singh, S. Gupta, and A. Ray, “Symbolic dynamic analysis of surface deformation during fatigue crack initiation,” *Measurement Science and Technology*, vol. 21, no. 4, p. 043003, 2010.

[89] V. Rajagopalan, S. Chakraborty, and A. Ray, “Estimation of slowly varying parameters in nonlinear systems via symbolic dynamic filtering,” *Signal Processing*, vol. 88, no. 2, pp. 339 – 348, 2008.

[90] S. Chakraborty, A. Ray, A. Subbu, and E. Keller, “Analytic signal space partitioning and symbolic dynamic filtering for degradation monitoring of electric motors,” *Signal, Image and Video Processing*, vol. 4, no. 4, pp. 399–403, 2010.

[91] P. Nomikos and J. F. MacGregor, “Multivariate spc charts for monitoring batch processes,” *Technometrics*, vol. 37, no. 1, pp. 41–59, 1995.

[92] A. Kaiser and T. Schreiber, “Information transfer in continuous processes,” *Physica D: Nonlinear Phenomena*, vol. 166, no. 1–2, pp. 43 – 62, 2002.

[93] J. Principe, “Information theoritic learing: Reny’s entropy and kernel perspective,” *Springer*, 2010.

[94] S. Sarkar, K. Mukherjee, X. Jin, D. Singh, and A. Ray, “Optimization of symbolic feature extraction for pattern classification,” *J. Signal processing*, vol. 92, no. 3, pp. 625–635, 2012.

[95] R. Steuer, L. Molgedey, W. Ebeling, J. Montao, and A. Miguel, “Entropy and optimal partition for data analysis,” *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 19, no. 2, pp. 265–269, 2001.

[96] S. Sarkar, S. Sarkar, N. Virani, A. Ray, and M. Yasar, “Sensor fusion for fault detection and classification in distributed physical processes,” *J. Frontiers in Robotics and AI*, vol. 1, p. 16, 2014.

[97] T. Schürmann and P. Grassberger, “Entropy estimation of symbol sequences,” *An Interdisciplinary Journal of Nonlinear Science*, vol. 6, no. 3, 1996.

[98] S. Sarkar, S. Sarkar, K. Mukherjee, A. Ray, and A. Srivastav, “Multi-sensor information fusion for fault detection in aircraft gas turbine engines,” *Proc. Inst. Mech. Eng., Part G: J. Aerospace Eng*, vol. 227, no. 12, pp. 1988–2001, 2013.

[99] S. Gupta and A. Ray, “Symbolic dynamic filtering for data-driven pattern recognition,” *Pattern recognition: theory and application*, pp. 17–71, 2007.

[100] K. Mukherjee and A. Ray, “State splitting and merging in probabilistic finite state automata for signal representation and analysis,” *J. Signal Processing*, vol. 104, pp. 105 – 119, 2014.

[101] X. Wang, A. Ray, and A. M. Khatkhate, “On-line identification of language measure parameters for discrete-event supervisory control,” *Applied Mathematical Modelling*, vol. 29, no. 6, pp. 597 – 613, 2005.

[102] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems.* Springer Science & Business Media, 2000.

[103] B. Rashidi, M. Esmaeilpoor, M. R. Homaeinezhad, and M. Zoghi, "Error based self-regulating pid angle control of variable structure redundant brushed dc motors," *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pp. 274–279, 2014.

[104] R. Abdullah, A. Hussain, K. Warwick, and A. Zayed, "Autonomous intelligent cruise control using a novel multiple-controller framework incorporating fuzzy-logic-based switching and tuning," *Neurocomputing*, vol. 71, no. 13, pp. 2727 – 2741, 2008.

[105] D. Fister, I. Fister, I. Fister, and R. Šafarič, "Parameter tuning of pid controller with reactive nature-inspired algorithms," *Robotics and Autonomous Systems*, vol. 84, pp. 64 – 75, 2016.

[106] M. S.-M. Edwin Lughofer, "Autonomous data stream clustering implementing split-and-merge concepts – towards a plug-and-play approach," *Information Sciences*, vol. 304, pp. 54 – 79, 2015.

[107] D. Deng and N. Kasabov, "On-line pattern analysis by evolving self-organizing maps," *Neurocomputing*, vol. 51, pp. 87 – 103, 2003.

[108] B. Fritzke, "A growing neural gas network learns topologies," *Adv. Neural Inform. Process. Syst*, vol. 7, pp. 625 – 632, 1995.

[109] J. W. G. Gan, C. Ma, "Data clustering: theory, algorithms, and applications," *SIAM, Society for Industrial and Applied Mathematics*, 2007.

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

[111] J. Peters, J. M. Mooij, D. Janzing, and B. Scolkopf, "Identifiability of causal graphs using functional models," *Proc. 27th Conf. on Uncertainty in Artificial Intelligence*, 2011.

[112] P. Jonas, D. Janzing, and B. Schölkopf, "Elements of causal inference: Foundations and learning algorithms," *Cambridge, MIT Press*, 2017.

[113] J. Pearl, "Causality: Models, reasoning, and inference," *Cambridge University Press*, vol. Second Edition, 2009.

[114] J. Pearl, "Probabilistic reasoning in intelligent systems," *Morgan Kaufmann Publications*, 1998.

[115] T.-D. Le, T. Hoang, J. Li, L. Liu, H. Liu, and S. Hu, “A fast pc algorithm for high dimensional causal discovery with multi-core pcs,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 5, pp. 1483–1495, 2019.

[116] V. A. Vakorin, O. A. Krakovska, and A. R. McIntosh, “Confounding effects of indirect connections on causality estimation,” *Journal of Neuroscience Methods*, vol. 184, no. 1, pp. 152 – 160, 2009.

[117] J. Demmel, I. Dumitriu, and O. Holtz, “Fast linear algebra is stable,” *Numerische Mathematik*, vol. 108, no. 1, pp. 59–91, 2007.

[118] K. Zhang, B. Huang, J. Zhang, C. Glymour, and B. Schölkopf, “Causal discovery from nonstationary/heterogeneous data: Skeleton estimation and orientation determination,” *IJCAI*, 2017.

[119] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to algorithms,” *MIT Press and McGraw-Hill*, no. 22.3, pp. 540–549, 2001.

[120] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Scholkopf, “Nonlinear causal discovery with additive noise models,” *Curran Associates, Inc.*, pp. 689–696, 2009.

[121] J. Downs and E. Vogel, “Industrial challenge problems in process control a plant-wide industrial process control problem,” *J. Comp. Chem. Eng*, vol. 17, no. 3, pp. 245 – 255, 1993.

[122] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, “Fisher discriminant analysis with kernels,” pp. 41–48, 1999.

[123] M. Pal, “Random forest classifier for remote sensing classification,” *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[124] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” *ACM*, pp. 1096–1103, 2008.

[125] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[126] R. Salakhutdinov and G. E. Hinton, “Deep boltzmann machines,” *In AISTATS*, p. 3, 2009.

[127] P. Sermanet, S. Chintala, and Y. LeCun, “Convolutional neural networks applied to house numbers digit classification,” *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3288–3291, Nov. 2012.

[128] B. Rashidi, M. Sundaram, and Q. Zhao, “Improved feature extraction and fault detection in non-stationary process through unsupervised machine learning,” *US. Patent, in filling process, Honeywell Process Solution*, vol. H0064385-0113, 2018.

[129] B. Rashidi and Q. Zhao, “Quality-related fault detection for processes with time-varying measurements,” *2019 18th European Control Conference (ECC)*, pp. 3873–3879, Jun. 2019.

Appendices

Algorithm 7 Principal component analysis (PCA) based on conducting SVD on the standardized data [23]

```
For training data $X \in \mathcal{R}^{N\times m}$, conduct mean centering and standardization.
perform singular value decomposition (SVD) $\frac{1}{\sqrt{N-1}}X = [\hat{U}\ \tilde{U}]\Lambda[\hat{V}\ \tilde{V}]^T$.
By following the cumulative percentage criteria choose the first $r$ columns of the matrix
    $U$ which include 95% (tuning parameter) of the variables variance.
Build the project matrices $M_{SPE} = \tilde{V}\tilde{V}^T$ and $M_{T^2} = \hat{U}\Lambda^{-0.5}\hat{U}^T$ from the loading vectors
    for generating the proper residual signals.
The upper control limit (UCL) for the Hotelling's $T^2$ statistic is calculated based on the
    fact that under normal operating condition, $T^2$ follows a $\mathcal{F}$ distribution as $UCL_{T^2} =$
    $(N-m)/(m(N-l))\mathcal{F}_\alpha(m, N-m)$.
The upper control limit (UCL) for the SPE index is calculated as $UCL_{SPE} =$
    $\theta_1(\frac{c_\alpha\sqrt{2\theta_2 h_0^2}}{\theta_1}+1+\frac{\theta_2 h_0(h_0-1)}{\theta_1^2})^{(1/h_0)}$, where $c_\alpha$ is the confidence interval that corresponds
    to the $1-\alpha$ percentile of the normal distribution. Also, $\theta_i = \sum_{j=m+1}^{r}\lambda_j^2,\ i = 1,2,3$ and
    $h_0 = 2\theta_1\theta_3/(3\theta_2^2)$.
The projection matrix for the combined index is $M_\phi = \frac{M_{SPE}}{UCL_{SPE}} + \frac{M_{T^2}}{UCL_{T^2}}$.
```

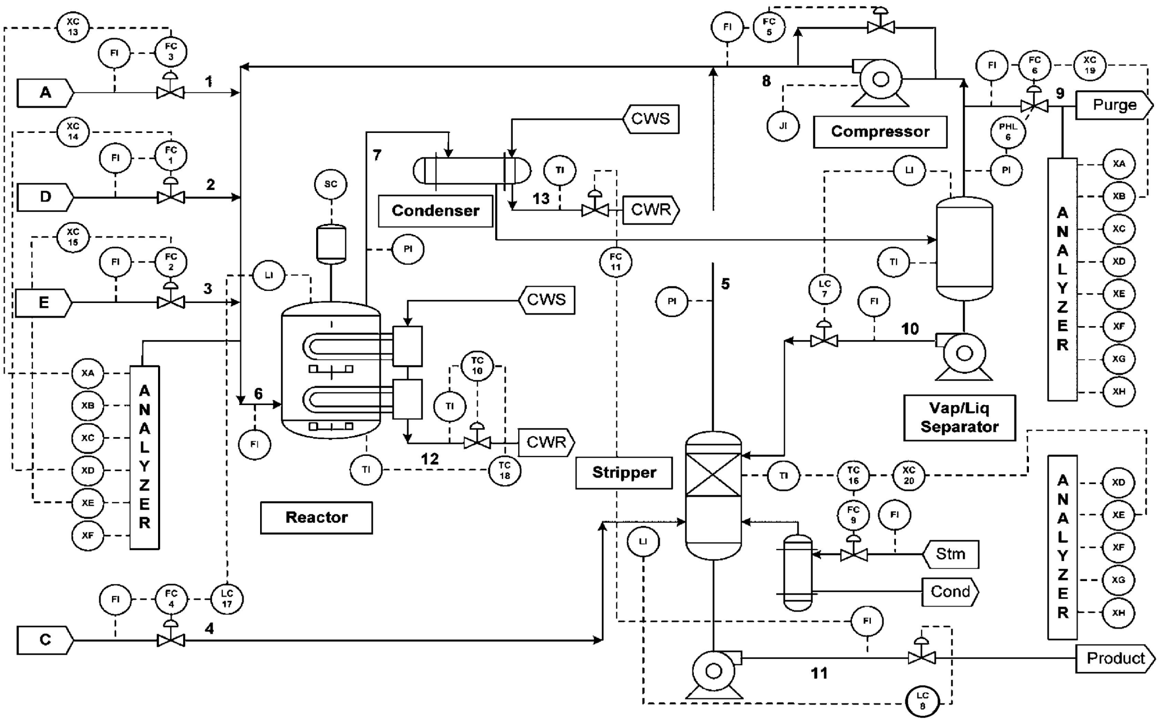


Figure 2: Tennessee Eastman benchmark process flowchart

Algorithm 8 kernel PCA algorithm for generating the residual signal ψ in non-linear processes [34]

```
• Training Phase:
Calculate uncentered training kernel matrix $\mathcal{K}$ using Gram kernel $\mathcal{K}_{ij} = \kappa(x(i), x(j))$.
Compute the centered kernel $K$ by using Eq. 4.1.
Conduct eigen-value decomposition on $K$ using Eq. 4.3.
Calculate the coordinates of $\Phi(X)$ as $Y = \Lambda_k^{1/2} U_k^T$ and determine the covariance matrix $C^Y = \frac{1}{N-1} Y^T Y$.
Apply SVD on $C^Y$ and find $\hat{U}$ and $\hat{\Lambda}$.
Compute upper control limit for $\psi$ residual by following procedure in [91].
• Testing Phase:
Calculate the uncentered kernel vector $\kappa(x^*)$ for testing vector $x^* \in \mathcal{R}^{(1\times m)}$.
Compute the centered kernel vector $k(x^*)$ using Eq. 4.2.
Obtain the reduced-coordinate $y = \Lambda_k^{-1/2} U_k^T k(x^*)$.
Calculate the residual $\psi(i) = (I_r - \hat{U}\hat{U}^T)y$.
In real-time calculation, if the $\psi(i) \geq UCL_{\psi}$, as it is shown in Fig. (4.1), the on/off
mechanism activates the root cause diagnosis mechanism to analysis the causality between
process variables and residual signal.
```

Table 1: Tennessee Eastman process fault

Index	Description	Index	Description
X1	A feed (stream 1)	X18	Stripper temperature
X2	D feed (stream 2)	X19	Stripper stream flow
X3	E feed (stream 3)	X20	Compressor work
X4	Total feed (stream 4)	X21	Reactor cooling water outlet temperature
X5	Recycle flow (stream 8)	X22	Condenser cooling water outlet temperature
X6	Reactor feed rate (stream 6)	X23	D feed flow (stream 2)
X7	Reactor pressure	X24	E feed flow (stream 3)
X8	Reactor level	X25	A feed flow (stream 1)
X9	Reactor temperature	X26	Total feed flow (stream 4)
X10	Purge rate (stream 9)	X27	Compressor recycle valve
X11	Separator temperature	X28	Purge valve (stream 9)
X12	Separator level	X29	Separator pot liquid flow (stream 10)
X13	Separator pressure	X30	Stripper liquid product flow
X14	Separator underflow (stream 10)	X31	Stripper steam valve
X15	Stripper level	X32	Reactor cooling water flow
X16	Stripper pressure	X33	Condenser cooling water flow
X17	Stripper underflow (stream 11)		

Table 3: Selected parameters for PFSAs construction of TEP variables

Index	ϵ_h	$\|\Sigma\|$	D	$\|Q\|$	Index	ϵ_h	$\|\Sigma\|$	D	$\|Q\|$
X1	0.15	5	2	19	X18	0.10	8	3	201
X2	0.15	3	2	9	X19	0.15	3	1	3
X3	0.15	3	2	9	X20	0.15	5	2	25
X4	0.15	4	1	9	X21	0.15	4	2	16
X5	0.15	4	2	12	X22	0.15	5	2	18
X6	0.15	4	2	16	X23	0.10	8	3	311
X7	0.15	4	2	14	X24	0.10	7	2	35
X8	0.15	4	1	4	X25	0.10	7	2	31
X9	0.15	5	2	25	X26	0.15	5	2	25
X10	0.10	4	3	43	X27	0.15	1	0	1
X11	0.10	5	2	25	X28	0.10	6	2	36
X12	0.15	4	2	10	X29	0.15	4	2	16
X13	0.15	5	2	24	X30	0.15	3	1	3
X14	0.15	3	1	3	X31	0.15	1	0	1
X15	0.15	4	2	16	X32	0.15	4	2	16
X16	0.10	5	3	41	X33	0.15	3	1	3
X17	0.15	3	1	3	ψ	0.15	7	3	220

Table 2: Tennessee Eastman process fault (the highlighted scenarios are considered for simulation case study).

Fault scenario	Process variables	Type
IDV(0)	Normal operation	Step
IDV(1)	A/C feed ratio, B composition constant	Step
IDV(2)	B composition, A/C ratio constant	Step
IDV(3)	D feed temperature	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	condenser cooling water inlet temperature	Step
IDV(6)	A feed loss	Step
IDV(7)	C header pressure loss-reduced availability	Step
IDV(8)	A,B,C feed composition	**Random variation**
IDV(9)	D feed temperature	Random variation
IDV(10)	C feed temperature	**Random variation**
IDV(11)	Reactor cooling water inlet temperature	**Random variation**
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Random variation
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking

Algorithm 9 (Helper Function): Proposed procedure to test a variable to be a source intermediate variable (SIV)

```
**Helper Function Inputs:** $P_s^{all}$, $P_t^{all}$
$P_1 \leftarrow empty$ (dummy potential variable)
$V_{node} \leftarrow empty$ (a list to store the visited nodes)
$max_s \leftarrow$length of the longest path in $P_s^{all}$
$max_t \leftarrow$length of the longest path in $P_t^{all}$
loop(1): $d = 0, ..., max_t - 1$
loop(2): $j_t = 0, ..., L_{P_t^{all}-1}$ (number of all path in $P_t^{all}$)
$len_t \leftarrow$ length of $P_t^{all}[j_t]$
if $d < len_t - 1$ then:
    $dp_t \leftarrow P_t^{all}[j_t][(len_t - d - 2) : (len_t - 1)]$ (dummy vector that gets the current path from the $d^{th}$ to one before last variable)
    $dv_t \leftarrow P_t^{all}[j_t][(len_t - d - 2)]$ (dummy variable gets the $d^{th}$ variable to the last of the path)
    if $dv_t \notin V_{node}$ then:
        $V_{node} \leftarrow V_{node} + dv_t$
        loop(3): $dd = 0, ..., max_s$
        loop(4): $j_s = 0, ..., L_{P_s^{all}}$ (number of all path in $P_s^{all}$)
        $len_s \leftarrow$ length of $P_s^{all}[j_s]$
        if $dd < len_s - 1$ then:
            $dp_s \leftarrow P_s^{all}[j_s][(len_s - dd - 2) : (len_s - 1)]$ (dummy vector that gets the current path from the $dd^{th}$ to one before last variable)
            $dv_s \leftarrow P_s^{all}[j_s][(len_s - dd - 2)]$ (dummy variable gets the $dd^{th}$ variable to the last of the path)
            if $dv_s == dv_t$ then:
                if $L_{dp_s} > 1$ and $dp_s \neq dp_t$ then:
                    $Cond_1 = True$
                    if Check if any of the vertices in $dp_s$ exist in $P_1$ then:
                        $Cond_1 = True$
                    $Cond_2 = True$
                    if Check if any of the vertices in $dp_t$ exist in $P_1$ then:
                        $Cond_2 = True$
                    if $dv_t \notin P_1$ and $\{Cond_1 or\ Cond_2\}$ then:
                        $P_1 \leftarrow P_1 + dv_t$ append $dv_t$ to $P_1$
                if $L_{dp_s} == 1$ then:
                    if $dv_t \notin P_1$ then:
                        $P_1 \leftarrow P_1 + dv_t$ append $dv_t$ to $P_1$
Output:$P_1 \Rightarrow$ validated SIVs
```