Key Performance Monitoring and Diagnosis in Industrial Automation Processes

Von der Fakultät für Ingenieurwissenschaften
Abteilung Elektrotechnik und Informationstechnik
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Haiyang Hao
aus
Shaanxi, V.R. China

1. Gutachter: Prof. Dr. -Ing. Steven X. Ding
2. Gutachterin: Prof. Dr. -Ing. habil. Ping Zhang
Tag der mündlichen Prüfung: 21.05.2014

Acknowledgements

This work was done during my PhD study in the Institute for Automatic Control and Complex Systems (AKS) at the University of Duisburg-Essen. I would like to express my deepest gratitude to Prof. Dr. -Ing Steven X. Ding, my honorable mentor, for his guidance to my scientific research work. My sincere appreciation must also go to Prof. Dr. -Ing. habil. Ping Zhang for her interest in my work. Her valuable comments improved the quality of this thesis.

I would like to thank my former group mates, Dr. Yin and Dr. Haghani, for all the discussions and cooperation during the EC and ZIM projects. My special thanks should go to Dr. Köppen-Seliger, who shared her valuable experiences on research projects and scientific writing. I would also like to thank Hao for his help and support during the last months of the EC project. Moreover, I would like to thank my new group mates, Kai and Zhiwen, for their kind help and all the very impressive discussions.

I would like to thank Linlin, Kai, Dr. Köppen-Seliger, Dr. Shardt, Hao and Zhiwen, for their review and suggestions to my dissertation. My thanks also go to all the AKS colleagues, Sabine, Eberhard, Klaus, Ulrich, Jonas, Tim K., Tim D., Chris, Christoph, Shane, Kristina, Ying, Dongmei, Minjia, Sihan, Changchen, Yong, Ping, Zhangming, Van Bien, Dr. Yuan, Dr. Yang, Prof. Peng, Prof. Huang and Prof. Lei. I cannot finish the work at this level without their help.

Finally, I would like to thank the European Commission (EC-FP7 project PAPYRUS, grant No. 257580) and AKS for the financial support during my research.

Duisburg, in June 2014 Haiyang Hao

To my parents, brother, and
my wife, Fei

Table of Contents

Nomenclature

Abbreviations

CSTH	Continuous Stirred Tank Heater
CW	Cold Water
DO	Diagnostic Observer
DPP	Distributed-Parameter Process
EU	European Union
EVD	EigenValue Decomposition
FDF	Fault Detection Filter
GLR	Generalized Likelihood Ratio
HW	Hot Water
ICA	Independent Component Analysis
I/O	Input/Output
KDE	Kernel Density Estimation
KL	Karhunen-Loève
KPI	Key Performance Indicator
LPP	Lumped-Parameter Process
LTI	Linear Time-Invariant
MIMO	Multiple-Input and Multiple-Output
MLE	Maximum Likelihood Estimate
NIPALS	Nonlinear Iterative PArtial Least Squares
ODE	Ordinary Differential Equation
OEE	Overall Equipment Effectiveness
PCA	Principal Component Analysis
PD	Paper Drying
PDE	Partial Differential Equation
PI	Proportional-Integral
PLS	Partial Least Squares
PS	Parity Space
RBC	Reconstruction-Based Contribution
SCADA	Supervisory Control And Data Acquisition

SVD	Singular Value Decomposition
TE	Tennessee Eastman

Mathematical symbols

$\forall$	for all
$\in$	belong to
$\subset$	subset
$\sim$	follow
$\approx$	approximately equal
$:=$	defined as
$>>$	much greater than
$\Longrightarrow$	imply
$\Longleftrightarrow$	equivalent to
x	a scalar
$\mathbf{x}$	a vector
$\mathbf{X}$	a matrix
$\mathcal{R}^n$	space of n-dimensional vectors
$\mathcal{R}^{m\times n}$	space of m by n matrices
$y(x)$	a function
$\mathbf{y}(x)$	a vector with functions as elements
$\mathbf{Y}(x)$	a matrix with functions as elements
$\mathcal{H}^n$	space of n-dimensional vectors with functions as elements
$\mathcal{H}^{m\times n}$	space of m by n matrices with functions as elements
$\mathcal{L}(\mathbf{z}(x,t))$	linear operator on $\mathbf{z}(x,t)$
$\mathbf{1}_N$	an N-dimensional column vector with all elements equal one
$\mathbf{I}_n$	an n by n identity matrix
$\|\|\mathbf{x}\|\|^2$	$:= \mathbf{x}^T\mathbf{x}$
$(m(x), n(x))$	$:= \int_\alpha^\beta m(x)n(x)\mathrm{d}x,\ x \in [\alpha, \beta]$
$\|\|m(x)\|\|^2$	$:= \int_\alpha^\beta m^2(x)\mathrm{d}x,\ x \in [\alpha, \beta]$
$\mathbf{X}(i:j, p:q)$	a submatrix consisting of the i-th to the j-th rows and the p-th to the q-th columns of $\mathbf{X}$
$\mathbf{X}^{-1}$	inverse of $\mathbf{X}$
$\mathbf{X}^\perp$	orthogonal complement of $\mathbf{X}$
$\mathbf{X}^+$	pseudoinverse of $\mathbf{X}$
$\mathbf{X}^T$	transpose of $\mathbf{X}$
$\mathrm{corr}(x, y)$	correlation coefficient between x and y

$\det(\mathbf{X})$	determinant of $\mathbf{X}$
$\text{diag}(\mathbf{x})$	a diagonal matrix formed with the elements of $\mathbf{x}$
$\text{diag}(\mathbf{X})$	a column vector formed with the diagonal elements of $\mathbf{X}$
$E(x)$ or $E(\mathbf{x})$	expected value of x or $\mathbf{x}$
$P(x)$ or $P(\mathbf{x})$	unconditional probability that an event, x or $\mathbf{x}$, occur
$\text{proj}_{\mathcal{P}}\, q$	projection of q onto the subspace $\mathcal{P}$
$\text{rand}(m, n)$	an m by n (uniform) random matrix
$\text{rank}(\mathbf{X})$	rank of $\mathbf{X}$
$\text{size}(\mathbf{X}, 1)$	number of rows of $\mathbf{X}$
$\text{std}(x)$ or $\text{std}(\mathbf{x})$	standard deviation of x or $\mathbf{x}$
$\text{svd}(\mathbf{X})$	SVD of $\mathbf{X}$
$\text{tr}(\mathbf{X})$	trace of $\mathbf{X}$
$\text{vec}(\mathbf{X})$	vectorization of $\mathbf{X}$
$\mathcal{F}(k, l)$	$\mathcal{F}$-distribution with k and l degrees of freedom
$\mathcal{F}_{1-\alpha}(k, l)$	critical value for $\mathcal{F}(k, l)$ for a significance level α
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\chi^2(k)$	chi-squared distribution with k degrees of freedom
$\chi^2_{1-\alpha}(k)$	critical value for $\chi^2(k)$ for a significance level α
$\mathcal{W}_m(k, \boldsymbol{\Sigma})$	Wishart distribution with k degrees of freedom based on the m-dimensional $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$

1 Introduction

1.1 Background and motivation of the work

The process industry has always played an important role in the European economy. Based on the statistics provided by the *Germany Trade & Invest* agency in 2011 [51], EU is the world's largest chemical exporter accounting for about 50 percent share of the global export market. Increasing global competition is setting even higher demands for the production safety, the product quality and the overall equipment effectiveness (OEE). To this end, the complexity and automation degree of industrial processes are significantly increasing with the technological developments. Today's production lines generally contain a great number of control loops with numerous embedded components like sensors, actuators [28]. However, a very important and complex task in process management, which is the response to the abnormal events, still remains largely a manual activity and is performed by the human operators. For difficult abnormal conditions, it should come as no surprise that human operators tend to make erroneous decisions and take actions which make matters even worse. Industrial statistics show that

- about 70% of the industrial accidents are caused by human errors [112].

Apart from the industrial accidents/disasters, those less serious abnormal events that decrease the OEE and lower the product quality can cause great economic losses, as an example,

- the petro-chemical industry loses 20 billion dollars per year due to process abnormalities [85].

From the available case studies provided by the *U.S. Chemical Safety Board*[1], a typical evolution of industrial accidents can be observed. As depicted in Figure 1.1, with the development of the abnormal events, the product quality, OEE and safety can be subsequently influenced. In practical processes, the performance of the low-level components degrades continuously along with the process running, *e.g.* fouling of the pipelines which

[1] http://www.csb.gov.

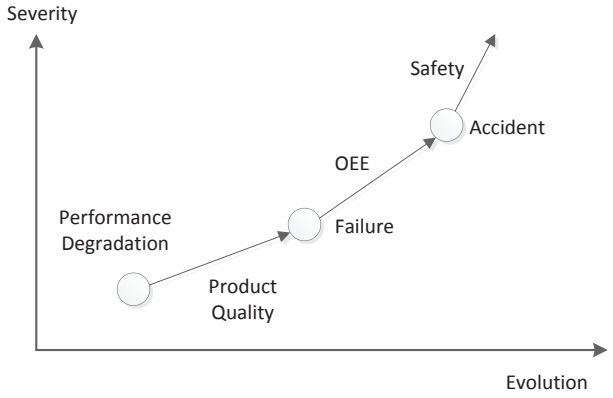


Figure 1.1: A typical evolution of the industrial accidents

reduces the flow rates, accumulation of solids in the chambers that decreases the heat transfer efficiency. If the decreasing performance is not monitored at an early stage and no maintenance action is performed, they can probably become component failure, *e.g.* cracks in the pipelines or chambers due to different factors of expansion and increased pressure. Finally, accidents can happen with the development of the failure, *e.g.* accumulated leaked flammable vapor can be ignited by the external activities.

Motivated by these observations, the concept of key performance indicator (KPI) has been introduced and the KPI-based process monitoring and fault diagnosis cover a wide range of requirements from industrial applications [28, 41, 86, 101, 113, 120, 137]. KPI aims at establishing a quantitative relationship between the performance of the low-level technical loops/components and the high-level product quality, OEE and process safety. To keep high enterprise profit, the KPI-based process monitoring and fault diagnosis tools play a very important role. During the past few decades, process monitoring and fault diagnosis techniques have received tremendous developments both in the research and practice, and are becoming an ingredient of modern automatic control system and often prescribed by authorities [24]. By amending the scope of process monitoring, huge losses can be averted [85]. In the next section, we attempt to review some of the major developments and progresses in the control community.

1.2 State of the art of techniques

1.2.1 Basic concept of process monitoring and fault diagnosis

The goal of process monitoring and fault diagnosis is not only to keep the plant operator and maintenance personnel better informed of the state of the process, but also to assist

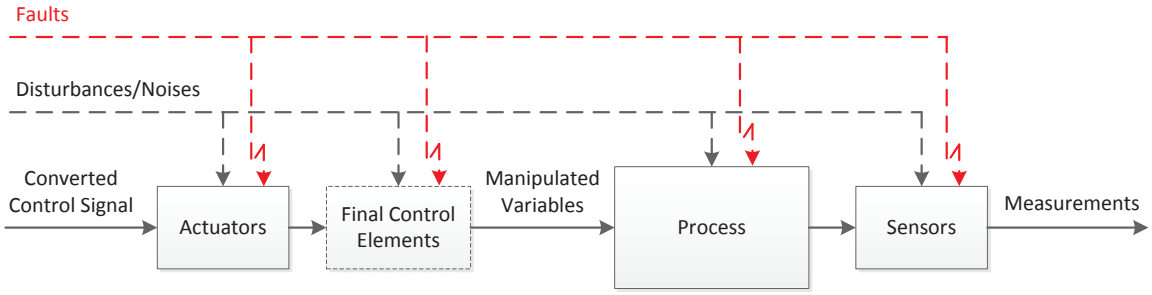


Figure 1.2: Representation of the industrial processes

them to make appropriate remedial actions to remove the abnormal behavior from the process [18]. In the control community, the abnormal behavior is generally defined as a fault, which is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition [53]. In the chemical and related process industry, *process monitoring* is widely accepted as a standard term for observing if any fault happens. In order to incorporate those performance degradation caused by non-faulty events, we introduce a more general term for the results achieved in this work, *i.e. performance monitoring*, which represents the detection of any unexpected behavior in the process. Once the performance monitoring system raises an alarm, it is very important to find out the root causes to prevent further losses. For this purpose, we adopt the traditional term *fault diagnosis*. In the data-driven framework, the main objective of a fault diagnosis system is to assist the plant operator and maintenance personnel to narrow down the investigation scope and thus shorten the downtime. For the consistency of this work, we use the term process monitoring when reviewing the existing techniques in the following.

1.2.2 Classification of process monitoring techniques

Figure 1.2 illustrates a typical configuration of the automatic control systems [57]. The control signal, typically 4-20 mA, needs to be converted to operate the actuators, which transform the control signal into the actions. Although not always necessary, the final control elements transmit the actuators' actions into a proper form to have the expected influence on the process. The outputs of actuators or final control elements are defined as the manipulated variables. The process is operated by these manipulated variables to make the raw materials into the final products. Sensors play a very important role in the modern automation industry, which not only provide the essential information for the process controllers but also are indispensable to the process monitoring systems.

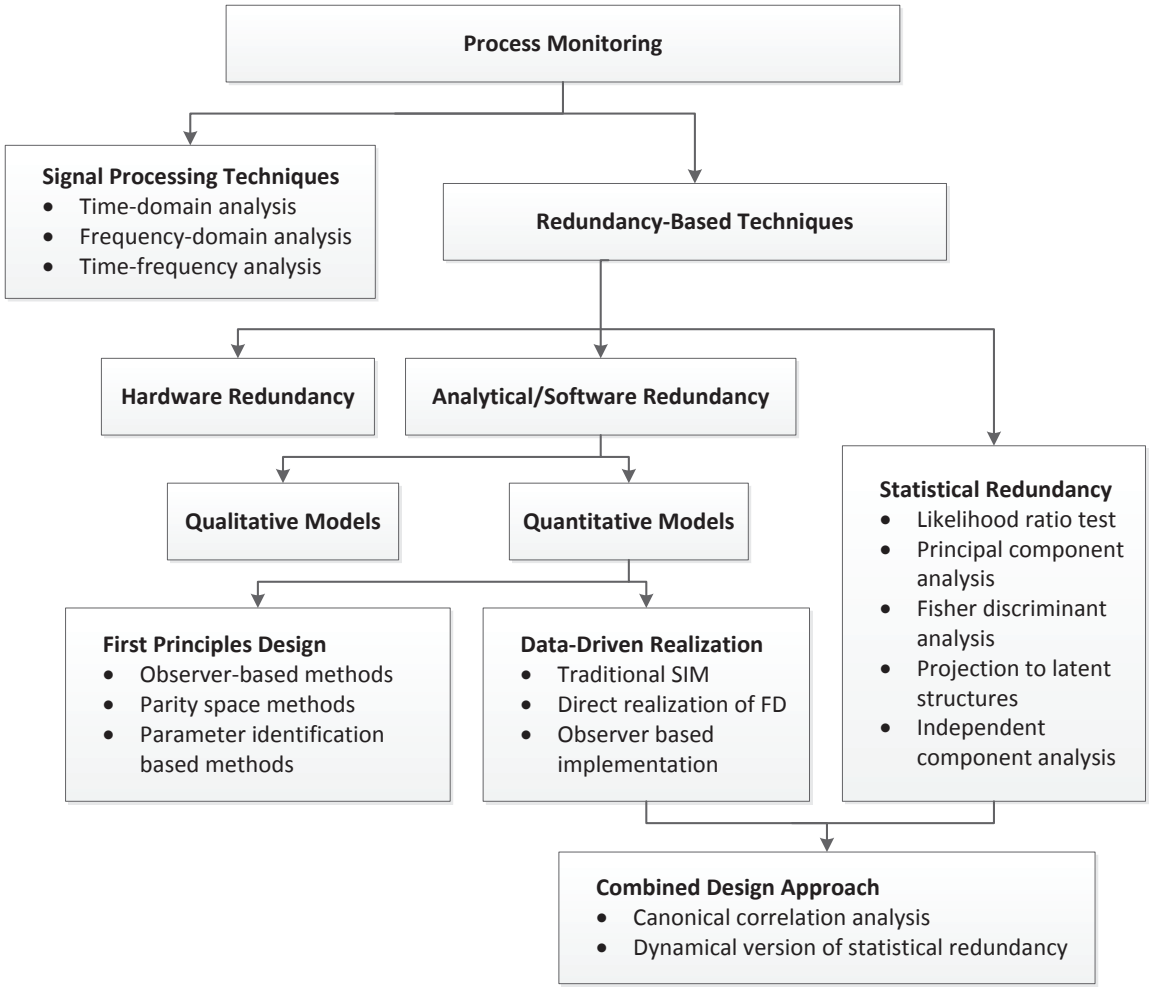


Figure 1.3: Classification of the process monitoring techniques

In the reality, all these elements are subject to various disturbances and noise that are always random. The primary task of process monitoring is to detect possible faults, *e.g.* actuator faults, process faults and sensor faults, that occur in the automation processes under such noisy environment based on the control signals and/or sensor measurements. Note that using the manipulated variables instead of the (converted) control signals may be an advantage for some fault detection schemes when the actuators are highly nonlinear, because then the required system equations do not contain the actuator nonlinearities [33].

A rough classification of the process monitoring techniques is given in Figure 1.3. The signal processing techniques have been well-established in the mechanical engineering community and remain very active. They use certain process signals carrying fault information of specific components, *e.g.* rotating machines, and have received great attention

in the practical applications [7, 53, 98, 123]. Differently, the redundancy-based techniques have been developed mainly in the control community. Based on some redundant knowledge of the process, the input and output signals are fed to the fault detection system to generate the residual signal that carries the fault information. To reveal the technical background for process monitoring, some traditional techniques are reviewed in the following.

Signal processing based techniques:

Traditional signal processing techniques include time- and frequency-domain analysis that are generally valid for linear stationary processes [72]. Those methods aim at extracting fault symptoms from the selected individual signals that carry rich information about the concerned components. The challenge lies in denoising the signals to recover useful fault features. Celebrated methods include the synchronous average and Fourier analysis [56]. Current research activities focus on time-frequency analysis that can be applied to nonlinear and non-stationary processes. Popular approaches include the wavelet analysis [99], stochastic resonance [71] and their enhancements [49, 106].

Hardware redundancy based techniques:

Redundancy is the key concept for the systematic process monitoring and has received great attention in the control community. Hardware redundancy is constructed by using identical hardware for the important components, especially in the nuclear, aviation and aerospace area where the safety issue is of the most importance [24, 82]. During the normal process operation, the system components and the redundant ones are fed with the same inputs, their outputs are continuously compared. Fault detection and isolation can be achieved by *e.g.* the "two out of three" principle. Hardware redundancy is of high reliability but with the price of the increased economical cost. Along with the developing technology and increasing automation degree in the processes, hardware redundancy may attract further interest for the process monitoring and especially for the fault tolerant control.

Qualitative model based techniques:

The qualitative model based techniques, which are also called knowledge-based methods, make use of the qualitative knowledge of the considered process (without the first principles) [35]. Different from the previously discussed approaches, they aim at finding the most possible root cause(s) by establishing the relationships of the fault symptoms, process states and measurements. Among them the causal analysis techniques, including the signed directed graphs [12, 62, 76, 119] and the fault/symptom tree analysis [70, 94, 105, 110], have been well established. These methods indicate the root causes

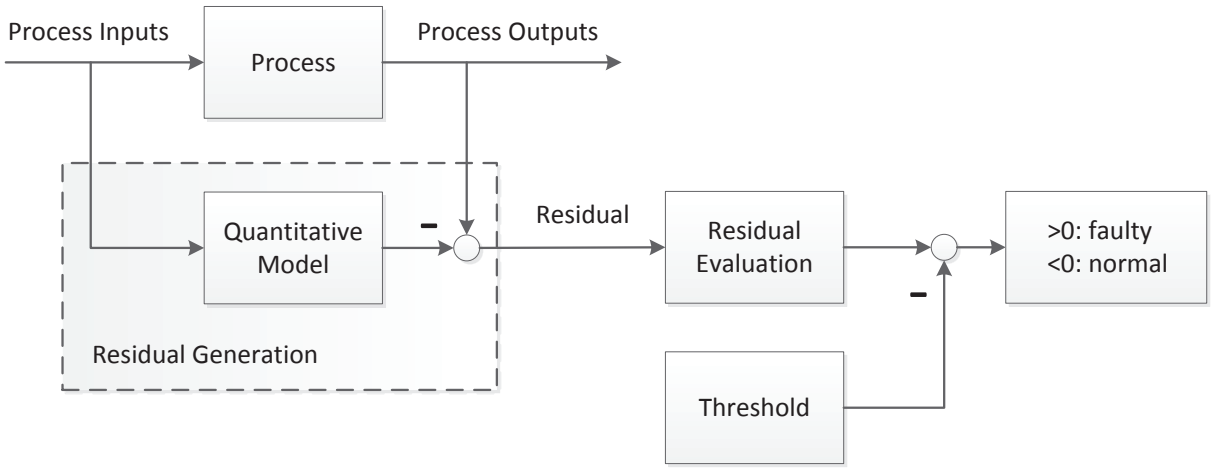


Figure 1.4: Traditional quantitative model based process monitoring scheme

by the logical analysis of all possible fault propagation paths. With the increasing scale and complexity of the processes, application of the causal analysis could be difficult and the performance may become unsatisfactory. Another kind of practically effective technique is the expert system [75, 111] that emulates the decision-making ability of a human expert. The knowledge base, which is the core of an expert system, is established by collecting adequate knowledge from the process experts. Expert systems are effective in the practical applications, especially for those mass-produced processes. In addition, pattern recognition [31, 96, 109] is receiving increasing research attention in the academic domain. When applied for the fault diagnosis, pattern recognition first establishes a set which consists of the features of the popular faults. Once a fault is reported, the fault feature is extracted online. By matching it with the feature set, the fault can be isolated.

Quantitative model based techniques:

Stimulated by the work of Beard [6] and Jones [59] in the early 70s, the quantitative model based process monitoring techniques have received tremendous developments in the control community. Figure 1.4 illustrates the basic idea behind this scheme. Different from the hardware redundancy based approaches, the redundancy (*i.e.* quantitative model) is generated in a much cheaper way, either based on the first principle dynamical modelling or data-driven identification/parameter estimation, and thus is defined as the analytical/software redundancy [24]. By comparing the outputs of the process and the model, the residual signals are generated. As described in Figure 1.2, real processes are contaminated by the disturbances, noise and faults. Nevertheless, the quantitative models only represent the uncontaminated dynamical part. As a result, it is clear that the residual signals contain all information about the faults. Due to the influence of the dis-

turbances/noise and the complexity of the generated residual signals, no easy decision can be made about the process status. In general, a residual evaluation function is designed to deliver a (non-zero) scalar signal. Moreover, a threshold needs to be determined based on the characteristic of the evaluation function as well as the property of the residual signals. Finally, if the evaluated residual crosses the threshold, then a fault is monitored; otherwise, the process is normal. In the academic domain, heavy research effort has been put on the design of the residual generators for the *lumped-parameter processes* (LPP) whose dynamics is governed by the ordinary differential equations (ODEs). These results are based on the first principle models [5, 10, 14, 24, 36, 40, 53, 89, 127, 133]. Well-known residual generation methods include the fault detection filter (FDF) [6, 15, 16, 25, 33, 59], diagnostic observer (DO) [24, 34, 118, 126], parity space (PS) approach [29, 79, 87, 88] and parameter identification based approaches [52].

In the practical applications, developing first principle models could be costly, time-consuming or even impossible. Thus data-driven techniques have attracted great attention both in the academy and industry. As the most popular approach, the subspace identification method has enjoyed tremendous development in the last 20 years [92] and been gradually applied in the process monitoring area [26, 74, 83, 124, 129]. Instead of deriving a state space model using the first principles, the system order as well as parameter matrices are directly identified from the noisy process input/output data. The identified system model is generally different from the real one subject to a regular state transformation. With this identified redundancy, traditional process monitoring techniques can be readily applied [67, 108, 134].

In the past 10 years, monitoring of the distributed parameter processes (DPPs), whose dynamics is described by partial differential equations (PDEs), started attracting some researchers' attention [22]. In those processes, the states, manipulated variables and process outputs are generally functions of spaces and of infinite dimension. DPPs widely exist in the industry, however, very little work can be found for the monitoring purpose and thus it is a quite new research direction. Due to the dimension of the states, techniques developed for the LPPs cannot be applied. To establish analytical redundancy, the states are decomposed into two subspaces corresponding to the slow and fast modes using the modal analysis technique. By neglecting the fast modes, a finite dimensional model is obtained. After lumping the distributed inputs/outputs into the slow subspace and feeding them to the finite dimensional model, the output error can be considered as residual signal for the monitoring purpose. The existing deterministic DPP monitoring approaches can be found in [3, 4, 32].

Statistical model based techniques:

For the large-scale processes, especially the chemistry-related ones, developing the previously mentioned qualitative- or quantitative model based monitoring techniques is not feasible mainly due to the complexity and high engineering effort. For these processes, the statistical monitoring techniques have been established and successfully applied in the practice [91]. Modern industrial processes are widely equipped with the SCADA (supervisory control and data acquisition) systems where all possible signals are measured and stored. The dimension of these measurements is huge and they follow certain statistical distributions. This observation has motivated the researchers to extract the statistical models from the huge amount of offline data. To monitor the actual process status, online data are fed into the statistical model to generate the residual signals. The most crucial challenge for such applications is how to effectively reduce the dimension of the data. Principal component analysis (PCA), originally developed in [48], is the most successful algorithm in the research domains like image processing and statistical process monitoring [13, 27, 37, 65, 116]. Projection to the latent structures, also known as partial least squares (PLS) in the sense of regression, is another popular statistical model based monitoring technique. Different from PCA which monitors the whole process variable space, PLS is able to monitor the quality-related and -unrelated process variable subspaces [64, 73, 78, 122, 136] and serves as an important tool for the online quality prediction [117]. For the residual evaluation, the multivariate control charts, *e.g.* Hotelling's T^2 and SPE, are widely applied. These test statistics are based on the assumption that the process disturbances/noise is multivariate normal distributed. In addition to PCA and PLS, the independent component analysis (ICA) is another statistical tool for statistical modelling [50]. ICA algorithm is able to extract the non-Gaussian sources driving a random process, based on which the test statistics are designed. It has been proven that ICA has better monitoring performance than PCA when the faults happened in the sources [43]. Further application of ICA for the process monitoring can be found in [1, 68, 69, 104, 125].

In addition, the combined data-driven quantitative and statistical redundancy is receiving growing interest for process monitoring. These approaches integrate both the information of the quantitative model and the statistical distribution of data thus outperform the individual approaches. Well-known approaches, *e.g.* the canonical correlation analysis and the dynamical version of the statistical redundancy based methods, can be sorted into this group [60, 66, 90, 100]. In practice, fault diagnosis of periodic systems is of great important since an early detection will prevent further damage to the process. The theory has been well established [130, 132] and special research interest has been focused on the influence of sampling period [128, 131]. Moreover, the artificial neural

network based fault diagnosis is another effective data-driven approach of great practical interest and has received comprehensive research interest [55, 61].

1.3 Objective of the work

The objective of this thesis is, based on the basic idea of the redundancy-based techniques, to design an efficient KPI monitoring and diagnosis system in a unified statistical framework. As described in the previous section, process monitoring techniques have been developed independently under different model descriptions, *i.e.* static model, lumped-parameter model and distributed-parameter model. However, their performance is quite restricted in the sense of algorithm efficiency, engineering effort and application scope. Thus the first task of this thesis is to

- develop improved residual generation schemes for KPI monitoring in static processes and LPPs with increased performance and lower engineering effort,
- propose novel (model-based and data-driven) residual generation schemes for KPI monitoring in DPPs with wide application scope and low engineering effort, and
- establish a statistical residual evaluation and decision making scheme.

In practice, since performance degradation is generally caused by multiplicative faults and its diagnosis is performed by the human experts, the work load is high and the efficiency is low. As a result, another task of this thesis is to develop an effective data-driven multiplicative fault diagnosis approach. This approach aims at identifying the root cause(s) of the performance degradation automatically, only using the process data. The purpose is to provide valuable information for the human experts to narrow down the investigation scope and thus to increase the OEE by reducing the downtime.

In addition to the theoretical contributions, the industrial application is another important objective of this thesis. The effectiveness of the proposed methods is demonstrated on three realistic industrial benchmark processes.

1.4 Outline of the thesis

The organization of this thesis is described in Figure 1.5. Following a general introduction to the process monitoring techniques given in Chapter 1, Chapter 2 first provides the technical descriptions of different industrial automation processes, *i.e.* static process, LPP and DPP. Then, the basic statistical process monitoring techniques like the generalized likelihood ratio (GLR) test, PCA and the standard PLS, are briefly summarized. In

addition, lumped parameter model based monitoring techniques are reviewed and their interconnections are addressed. Finally, a brief summary of the process monitoring techniques for the DPPs is given.

In Chapter 3, an alternative KPI monitoring scheme is proposed. Different from the standard approaches which provide low monitoring performance and involve complex computations, the KPI-related test statistic of the alternative scheme only monitors the process variable subspace that is related to the KPI. Two test statistics are designed and both the computation cost and the engineering effort are low. The performance improvement of the alternative scheme is demonstrated using numerical examples.

Chapter 4 presents a data-driven dynamic process monitoring approach using the subspace identification method. Based on the process and KPI data, an extended kernel representation of the process is identified with enhanced denoising performance. Different from the standard approaches, the covariance matrix of the residual vector is directly obtained in the identification step. Numerical examples are used to show the monitoring performance. In addition, this chapter plays an essential role for the data-driven realization of the monitoring techniques developed for the DPPs in the next chapter.

Chapter 5 mainly discusses the monitoring issues related to the DPPs. The concept of projection in the infinite dimensional space builds the basis of this chapter. Different from the existing work done in the deterministic framework, we propose a new efficient monitoring algorithm in the statistical framework. The achieved monitoring system is a kernel representation of the DPP. To deal with the projection error, a residual evaluation scheme is established for the stochastic processes with deterministic errors. To realize the proposed scheme in the data-driven framework, an efficient feature extraction method is used to identify the optimal basis functions. The effectiveness of the proposed methods is demonstrated using numerical examples.

In Chapter 6, a novel data-driven multiplicative fault diagnosis method is proposed. Different from the existing additive fault diagnosis approaches, the proposed method aims at diagnosing the more common multiplicative faults. Numerical examples are finally used to show its performance.

In Chapter 7, the algorithms developed in Chapters 3-6 are tested on three realistic industrial benchmark processes. Typical fault episodes are considered in different case studies. The test results show that the proposed methods are very suitable for practical applications.

Chapter 8 concludes the thesis and discusses the future scope.

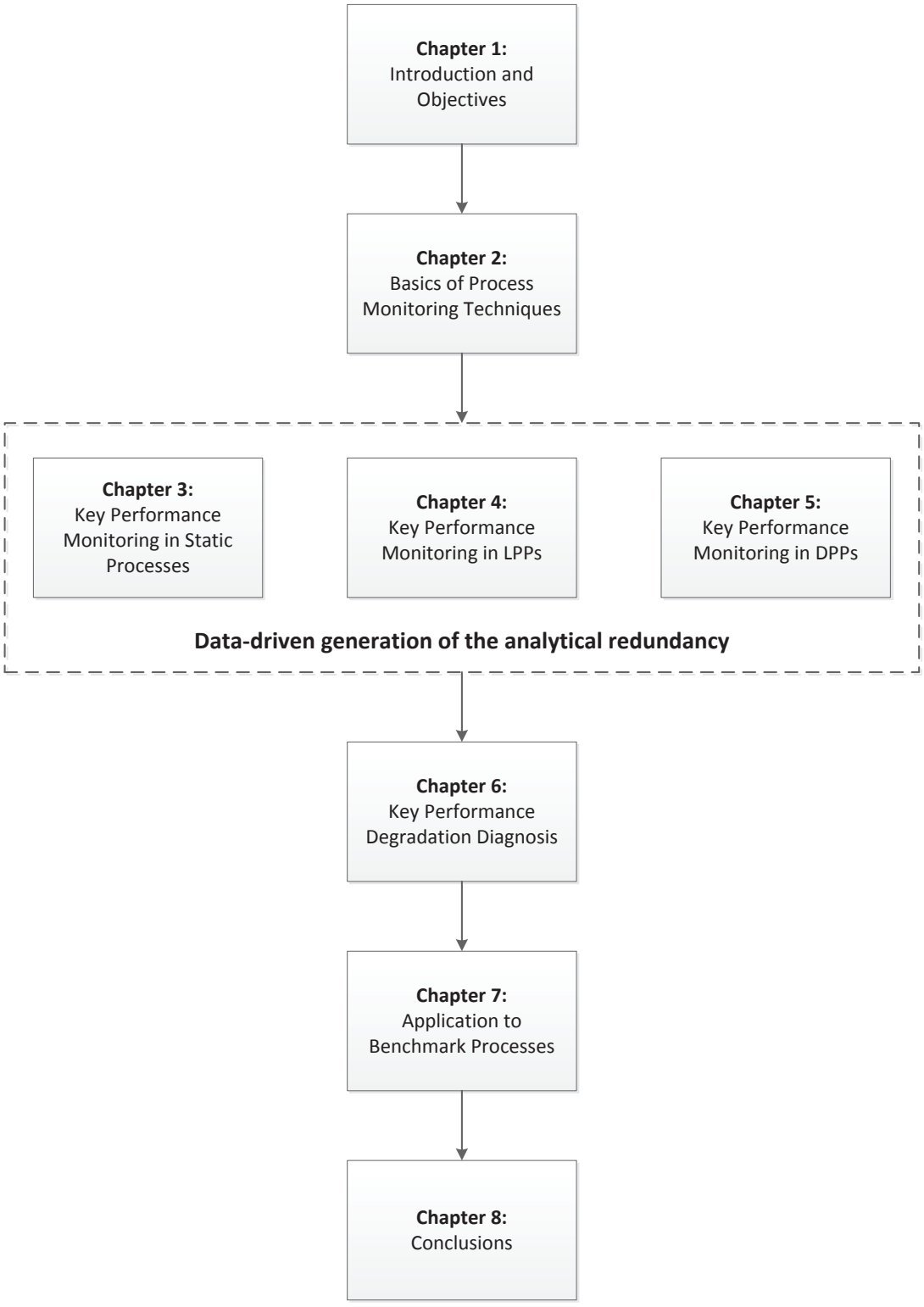


Figure 1.5: Organization of the chapters

2 Basics of process monitoring techniques

As discussed in the introduction, *residual* is the key concept of modern process monitoring systems. By comparing the properly evaluated residual signals with a predefined threshold, the performance/status of the automation processes can be timely reported. In order to generate the residual signals, *redundancy* is essential and serves as the core of a residual generator. The way how the redundancy is achieved greatly determines the performance of an established residual generator. Depending on the representation of automation processes, redundancy can be generated using the analytical quantitative models or the statistical distributions of the process data. Based on the mathematical descriptions of the automation processes, the objective of the chapter is to provide the basics of these two types of redundancy generation schemes which serve as the fundamental of this thesis.

2.1 Mathematical description of automation processes

2.1.1 Representation of static processes

For those (slow) processes which are in steady state, the low-level process variables and the high-level KPIs are expected to be stationary under normal production condition. As shown in Figure 2.1, their variations are only subject to the random factors, which can be represented by an algebraic equation as

$$\boldsymbol{\theta}_{obs}(k) = \boldsymbol{\Psi}_{obs}\mathbf{y}_{obs}(k) + \mathbf{b}_{obs} + \mathbf{e}_{\boldsymbol{\theta}_{obs}}(k) \tag{2.1}$$

where $\mathbf{y}_{obs} \in \mathcal{R}^m, \boldsymbol{\theta}_{obs} \in \mathcal{R}^l$ and $\mathbf{e}_{\boldsymbol{\theta}_{obs}} \in \mathcal{R}^l$ denote the multivariate normally distributed vectors of the low-level process variables, the high-level KPIs and the zero-mean noise which is uncorrelated with $\mathbf{y}_{obs}$ (*i.e.* $E(\mathbf{e}_{\boldsymbol{\theta}_{obs}}(\mathbf{y}_{obs} - E(\mathbf{y}_{obs}))^T) = 0$), respectively. The system parameters $\boldsymbol{\Psi}_{obs} \in \mathcal{R}^{l\times m}$ and $\mathbf{b}_{obs} \in \mathcal{R}^l$ are time-invariant.

During the fault-free operations, denote $\mathbf{y}_{obs}(k) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}_{obs}}, \boldsymbol{\Sigma}_{\mathbf{y}_{obs}})$ and $\boldsymbol{\theta}_{obs}(k) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}_{obs}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{obs}})$, $\forall\ k = 1, 2, \cdots, \infty$. The potential faults could influence both $\mathbf{y}_{obs}$ and

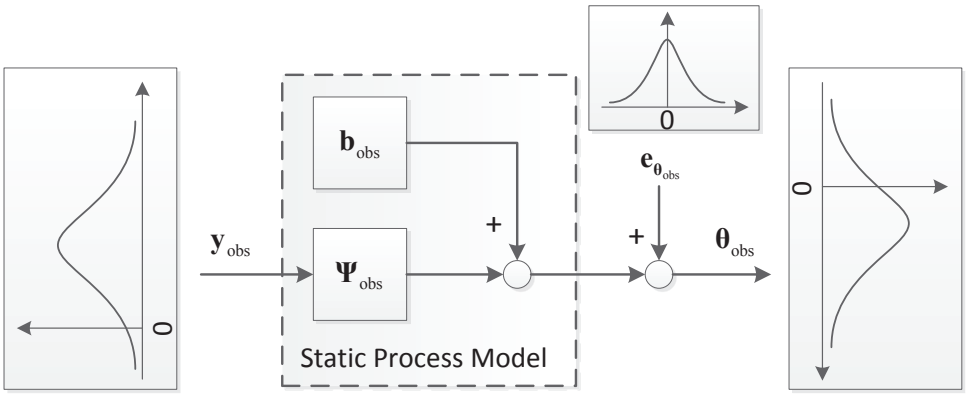


Figure 2.1: Illustration of static processes

$\boldsymbol{\theta}_{obs}$. Before introducing the fault models, the potential faults are divided into two categories in this study as

- additive faults: the faults that only influence the first order statistics, *i.e.* $\boldsymbol{\mu}_{\mathbf{y}_{obs}}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}_{obs}}$, and
- multiplicative faults: the faults that change the second order statistics. *i.e.* $\boldsymbol{\Sigma}_{\mathbf{y}_{obs}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{obs}}$.

In the reality, both kinds of faults could occur simultaneously. In the data-driven framework, the additive faults can be modelled as

$$\mathbf{y}_{obs,f}(k) = \mathbf{y}_{obs}(k) + \mathbf{f}_{y,obs}(k),\ \boldsymbol{\theta}_{obs,f}(k) = \boldsymbol{\theta}_{obs}(k) + \mathbf{f}_{\boldsymbol{\theta},obs}(k)$$

where $\mathbf{f}_{y,obs}(k)$ and $\mathbf{f}_{\boldsymbol{\theta},obs}$ denote those faults that influence the low-level process variables and the high-level KPIs, respectively. It is obvious that the former type of faults can propagate into the KPIs and thus change both $\mathbf{y}_{obs}$ and $\boldsymbol{\theta}_{obs}$.

In practice, malfunctions in the process often cause changes in the model parameters $\boldsymbol{\Psi}_{obs}$ and $\mathbf{b}_{obs}$, which could influence the second order statistics. These multiplicative faults can be modelled as

$$\mathbf{y}_{obs,mf}(k) = \mathbf{F}_{\mathbf{y}}(\mathbf{y}_{obs}(k) - \boldsymbol{\mu}_{\mathbf{y}_{obs}}) + \boldsymbol{\mu}_{\mathbf{y}_{obs}},\ \boldsymbol{\theta}_{obs,mf}(k) = \mathbf{F}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{obs}(k) - \boldsymbol{\mu}_{\boldsymbol{\theta}_{obs}}) + \boldsymbol{\mu}_{\boldsymbol{\theta}_{obs}}$$

where $\mathbf{F}_{\mathbf{y}}$ and $\mathbf{F}_{\boldsymbol{\theta}}$ are full-rank matrices describing the influences of multiplicative fault.

2.1.2 Representation of lumped-parameter processes

For processes exhibiting strong (temporal) dynamics with lumped states (*i.e.* LPP), ODE serves as a powerful tool for modelling and design of the monitoring systems. Among

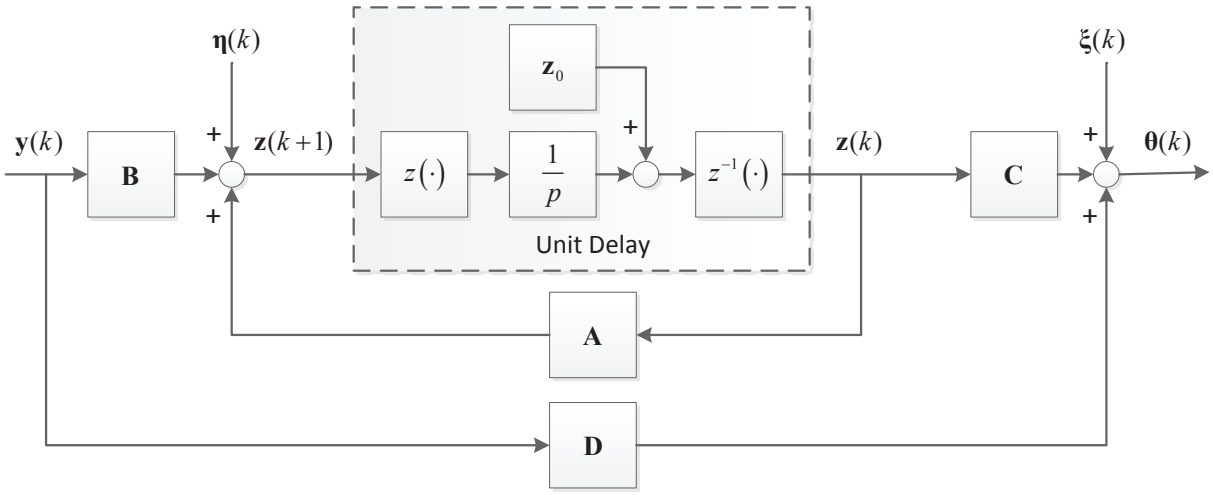


Figure 2.2: Block diagram of lumped-parameter systems

various ODEs, the linear time-invariant (LTI) one is widely used both in the theoretical study and the industrial applications. As shown in Figure 2.2, the standard form of the state space representation of a discrete-time LTI system is given by

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}\mathbf{y}(k) + \boldsymbol{\eta}(k),\ \mathbf{z}(0) = \mathbf{z}_0, \tag{2.2}$$

$$\boldsymbol{\theta}(k) = \mathbf{C}\mathbf{z}(k) + \mathbf{D}\mathbf{y}(k) + \boldsymbol{\xi}(k) \tag{2.3}$$

where $\mathbf{z} \in \mathcal{R}^n$ is called the state vector, $\mathbf{z}_0$ is the initial condition of the system, in this thesis, we assume that $\mathbf{y}_{obs} \in \mathcal{R}^m$ represents the low-level process vector and $\boldsymbol{\theta}_{obs} \in \mathcal{R}^l$ represents the high-level KPI vector, $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are uncorrelated white noise sequences that are uncorrelated with $\mathbf{z}_0$. $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ are appropriately dimensioned real constant matrices.

The faults in the LTI systems could be modelled in several ways. One widely adopted model is achieved by extending the model (2.2-2.3) to

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}\mathbf{y}(k) + \boldsymbol{\eta}(k) + \mathbf{E}\mathbf{f}(k),\ \mathbf{z}(0) = \mathbf{z}_0, \tag{2.4}$$

$$\boldsymbol{\theta}(k) = \mathbf{C}\mathbf{z}(k) + \mathbf{D}\mathbf{y}(k) + \boldsymbol{\xi}(k) + \mathbf{F}\mathbf{f}(k) \tag{2.5}$$

where $\mathbf{f} \in \mathcal{R}^{n_\mathbf{f}}$ is an unknown vector that represents all possible additive faults. Matrices $\mathbf{E}$ and $\mathbf{F}$ are of appropriate dimensions and describe how the faults influence the system.

The model (2.4-2.5) is a general representation of

- the faults in the KPI space by choosing $\mathbf{E} = \mathbf{0}$ & $\mathbf{F} = \mathbf{I}$, that is

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}\mathbf{y}(k) + \boldsymbol{\eta}(k),\ \boldsymbol{\theta}(k) = \mathbf{C}\mathbf{z}(k) + \mathbf{D}\mathbf{y}(k) + \boldsymbol{\xi}(k) + \mathbf{f}(k)$$

- or the faults in the process variable space by choosing $\mathbf{E} = \mathbf{B}$ & $\mathbf{F} = \mathbf{D}$, *i.e.*

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}(\mathbf{y}(k) + \mathbf{f}(k)) + \boldsymbol{\eta}(k),\ \boldsymbol{\theta}(k) = \mathbf{C}\mathbf{z}(k) + \mathbf{D}(\mathbf{y}(k) + \mathbf{f}(k)) + \boldsymbol{\xi}(k)$$

Although some process faults might be represented by (2.4-2.5) with appropriate $\mathbf{E}$ and $\mathbf{F}$, it is often the case that process faults in practice change the system parameters $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$. Systematic description of these process faults is given as

$$\begin{aligned}\mathbf{z}(k+1) &= (\mathbf{A} + \Delta\mathbf{A})\mathbf{z}(k) + (\mathbf{B} + \Delta\mathbf{B})\mathbf{y}(k) + \boldsymbol{\eta}(k),\ \mathbf{z}(0) = \mathbf{z}_0,\\ \boldsymbol{\theta}(k) &= (\mathbf{C} + \Delta\mathbf{C})\mathbf{z}(k) + (\mathbf{D} + \Delta\mathbf{D})\mathbf{y}(k) + \boldsymbol{\xi}(k)\end{aligned}$$

where $\Delta\mathbf{A}, \Delta\mathbf{B}, \Delta\mathbf{C}$ and $\Delta\mathbf{D}$ represent the influences of faults on the system parameters. Based on our early work in [44], these process faults could influence the second order statistics of the output data thus belong to multiplicative faults (the faults described by (2.4-2.5) only changes the first order statistics of the output data [44]).

2.1.3 Representation of distributed-parameter processes

In the large-scale industrial production industry, DPPs widely exist [19, 95]. The dynamics of the DPPs is related to both the time and the space, which is generally described by PDEs. Motivated by the description in [135], the general model of an n-th order DPP is given as

$$\begin{aligned}\mathcal{F}(\mathbf{y}(x,t), x, t, \mathbf{z}(x,t), \frac{\partial\mathbf{z}(x,t)}{\partial x}, \frac{\partial\mathbf{z}(x,t)}{\partial t}, \frac{\partial^2\mathbf{z}(x,t)}{\partial x^2}, \frac{\partial^2\mathbf{z}(x,t)}{\partial t\partial x}, \frac{\partial^2\mathbf{z}(x,t)}{\partial t^2}, \cdots,\\ \frac{\partial^n\mathbf{z}(x,t)}{\partial x^n}, \cdots, \frac{\partial^n\mathbf{z}(x,t)}{\partial t^n} + \boldsymbol{\eta}(x,t)) = 0,\\ \boldsymbol{\theta}(x,t) = \mathcal{G}\left(\mathbf{z}(x,t), \frac{\partial\mathbf{z}(x,t)}{\partial x}, \frac{\partial\mathbf{z}(x,t)}{\partial t}, \cdots\right) + \boldsymbol{\xi}(x,t)\end{aligned}$$

where $x \in [\alpha, \beta]$ denotes the spatial variable, t denotes the time, $\mathbf{y}(x,t)$ represents the spatially distributed low-level process variables, $\mathbf{z}(x,t)$ represents the spatially distributed states, $\boldsymbol{\eta}(x,t)$ is the distributed process noise, $\boldsymbol{\theta}(x,t)$ is the KPI measurements and $\boldsymbol{\xi}(x,t)$ denotes the measurement noise. To provide a rigid description of the DPPs, suitable initial and boundary conditions need to be supplemented.

The above description is much too complex for the real applications since the nonlinearity, time-varying and high order properties are involved. Considering the fact that the majority of processes arising in science and engineering can be well described by first

or second order PDEs [39], the following representation of DPPs is addressed in this thesis

$$\begin{aligned}\mathbf{A}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial t^2}+\mathbf{B}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial t\partial x}+\mathbf{C}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial x^2}+\mathbf{D}(x)\frac{\partial\mathbf{z}(x,t)}{\partial t}\\ +\mathbf{E}(x)\frac{\partial\mathbf{z}(x,t)}{\partial x}+\mathbf{F}(x)\mathbf{z}(x,t)+\mathbf{G}(x)\mathbf{y}(x,t)+\boldsymbol{\eta}(x,t)=0\\ \boldsymbol{\theta}(x,t)=\int_\alpha^\beta\mathbf{H}(x,x')\mathbf{z}(x',t)\mathrm{d}x'+\int_\alpha^\beta\mathbf{I}(x,x')\frac{\partial\mathbf{z}(x',t)}{\partial t}\mathrm{d}x'+\boldsymbol{\xi}(x,t)\end{aligned}$$

where $\mathbf{A}(x),\mathbf{B}(x),\mathbf{C}(x),\mathbf{D}(x),\mathbf{E}(x),\mathbf{F}(x),\mathbf{G}(x),\mathbf{H}(x,x')$ and $\mathbf{I}(x,x')$ are model parameter matrices whose elements are space functions. More detailed description and analysis of the above model can be found in Chapter 5.

Similar to the lumped-parameter cases, two kinds of faults can be modelled. The first one is additive and changes the mean value of the process and KPI data, which can be formulated as

$$\begin{aligned}\mathbf{y}_f(x,t)&=\mathbf{y}(x,t)+\mathbf{f}_\mathbf{y}(x,t)\\ \boldsymbol{\theta}_\mathbf{f}(x,t)&=\boldsymbol{\theta}(x,t)+\mathbf{f}_{\boldsymbol{\theta}}(x,t)\end{aligned}$$

where $\mathbf{f}_\mathbf{y}(x,t)$ and $\mathbf{f}_{\boldsymbol{\theta}}(x,t)$ denote the fault magnitudes. The second one happens in the distributed parameter matrices and can be modelled as

$$\mathbf{Q}_\mathbf{f}(x)=\mathbf{Q}(x)+\boldsymbol{\Delta}\mathbf{Q}(x)$$

where $\mathbf{Q}(x)$ represents a parameter matrix of the second order PDE and $\boldsymbol{\Delta}\mathbf{Q}(x)$ denotes the fault.

2.2 Basic monitoring techniques for static processes

In this section, basic statistical techniques for the monitoring of large-scale static processes are reviewed. From the statistical viewpoint, the model (2.1) is equivalent to

$$\boldsymbol{\theta}=\boldsymbol{\Psi}\mathbf{y}+\mathbf{e}_{\boldsymbol{\theta}} \tag{2.6}$$

where $\mathbf{y}$ and $\boldsymbol{\theta}$ are the normalized process variables and the KPIs, respectively

$$\mathbf{y}=\mathrm{diag}(\mathrm{std}(\mathbf{y}_{obs}^T))(\mathbf{y}_{obs}-E(\mathbf{y}_{obs})), \tag{2.7}$$

$$\boldsymbol{\theta}=\mathrm{diag}(\mathrm{std}(\boldsymbol{\theta}_{obs}^T))(\boldsymbol{\theta}_{obs}-E(\boldsymbol{\theta}_{obs})) \tag{2.8}$$

with $E(\cdot)$ denoting a column vector of the mean values and std$(\cdot)$ a column vector of the standard deviation of " $\cdot$ ". The parameters of the model (2.1) are connected to the one from (2.6) as

$$\begin{aligned}\boldsymbol{\Psi}_{obs} &= \operatorname{diag}(\operatorname{std}(\boldsymbol{\theta}_{obs}))\boldsymbol{\Psi}\operatorname{diag}(\operatorname{std}(\mathbf{y}_{obs})),\\ \mathbf{b}_{obs} &= E(\boldsymbol{\theta}_{obs}) - \boldsymbol{\Psi}_{obs}E(\mathbf{y}_{obs}).\end{aligned}$$

Although everywhere utilized while much less emphasized, the normalization procedure actually plays a central role for monitoring of those static processes. For applying these techniques where the process variables and the KPIs are not distinguished, the following organization of data is adopted

$$\mathbf{z} = \begin{bmatrix}\mathbf{y}\\ \boldsymbol{\theta}\end{bmatrix} \in \mathcal{R}^{n_\mathbf{z}},\ n_\mathbf{z} = m + l.$$

In the following subsections, the static model (2.6) instead of (2.1) is used as a starting point for introducing the monitoring techniques.

2.2.1 Generalized likelihood ratio test

Given the following model

$$\mathbf{z} = \mathbf{z}^* + \mathbf{f}$$

where $\mathbf{z}^* \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ represents the statistic feature of the normal process and $\mathbf{f}$ denotes a possible fault. The task is to test the following hypotheses based on the observation data $\mathbf{z}$ [5]

$$\begin{aligned}H_0,\ \text{null hypothesis: } \mathbf{f} = \mathbf{0},\ \text{fault-free},\\ H_1,\ \text{alternative hypothesis: } \mathbf{f} \neq \mathbf{0},\ \text{faulty}.\end{aligned}$$

The probability density functions of $\mathbf{z}^*$ and $\mathbf{z}$ are given as

$$P_0(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^{n_\mathbf{z}}\det(\boldsymbol{\Sigma})}}\,\mathrm{e}^{-\frac{1}{2}\mathbf{z}^T\boldsymbol{\Sigma}^{-1}\mathbf{z}}, \tag{2.9}$$

$$P_1(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^{n_\mathbf{z}}\det(\boldsymbol{\Sigma})}}\,\mathrm{e}^{-\frac{1}{2}(\mathbf{z}-E(\mathbf{z}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}-E(\mathbf{z}))}. \tag{2.10}$$

The log likelihood ratio is defined to be

$$s(\mathbf{z}) = 2\ln\frac{P_1(\mathbf{z})}{P_0(\mathbf{z})} = \mathbf{z}^T\boldsymbol{\Sigma}^{-1}\mathbf{z} - (\mathbf{z}-E(\mathbf{z}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}-E(\mathbf{z})). \tag{2.11}$$

Algorithm 2.1. *Model-based GLR test*

Based on $\mathbf{\Sigma}$ *and the online data* $\mathbf{z}_k,\ k = 1, \cdots, N,$

S1: Determine $\chi^2_{1-\alpha}(n_{\mathbf{z}})$ *using the* χ^2*-distribution with* $n_{\mathbf{z}}$ *degrees of freedom for a significance level* α.

S2: Set the threshold: $J_{GLR,th} = \chi^2_{1-\alpha}(n_{\mathbf{z}})$.

S3: Build $\bar{\mathbf{z}}$ *based on* $\mathbf{z}_k,\ k = 1, \cdots, N,$ *as* $\bar{\mathbf{z}} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{z}_k$.

S4: Construct the test statistic: $J_{GLR} = N\bar{\mathbf{z}}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{z}}$.

S5: Check the decision logic:

$$\begin{cases} J_{GLR} > J_{GLR,th} \Rightarrow & H_1 : \textit{faulty} \\ J_{GLR} \leq J_{GLR,th} \Rightarrow & H_0 : \textit{fault-free}. \end{cases}$$

To increase the confidence of the decision-making procedure, generally more samples are required. Assume that N samples of data, *i.e.* $\mathbf{z}_k,\ k = 1, \cdots, N$, are available, the above definition is extended as

$$\begin{aligned} S_1^N &= \sum_{k=1}^{N} 2\ln\frac{P_1(\mathbf{z}_k)}{P_0(\mathbf{z}_k)} \\ &= \sum_{k=1}^{N} \mathbf{z}_k^T\mathbf{\Sigma}^{-1}\mathbf{z}_k - \sum_{k=1}^{N}(\mathbf{z}_k - E(\mathbf{z}))^T\mathbf{\Sigma}^{-1}(\mathbf{z}_k - E(\mathbf{z})) \\ &= 2N\bar{\mathbf{z}}^T\mathbf{\Sigma}^{-1}E(\mathbf{z}) - NE(\mathbf{z})^T\mathbf{\Sigma}^{-1}E(\mathbf{z}),\ \bar{\mathbf{z}} := \frac{1}{N}\sum_{k=1}^{N}\mathbf{z}_k \\ &= N(\bar{\mathbf{z}}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{z}} - (\bar{\mathbf{z}} - E(\mathbf{z}))^T\mathbf{\Sigma}^{-1}(\bar{\mathbf{z}} - E(\mathbf{z}))) \end{aligned}$$

It is obvious that the maximum of S_1^N is $N\bar{\mathbf{z}}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{z}}$, which is achieved when $\bar{\mathbf{z}} = E(\mathbf{z})$. Since $E(\mathbf{z}) = \mathbf{f}$ is generally unknown, it is always replaced by the maximum likelihood estimate $\sum_{k=1}^{N}\mathbf{z}_k/N$, which is known as the GLR test. The maximum likelihood ratio is used as a test statistic as

$$J_{GLR} = N\bar{\mathbf{z}}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{z}}.$$

In the case that $\mathbf{\Sigma}$ is known *a priori*, since

$$\bar{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}, \frac{1}{N}\mathbf{\Sigma}) \Rightarrow N\bar{\mathbf{z}}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{z}} \sim \chi^2(n_{\mathbf{z}}),$$

the threshold of J_{GLR} is set as

$$J_{GLR,th} = \chi^2_{1-\alpha}(n_{\mathbf{z}})$$

where α denotes the significance level.

The complete procedures of GLR test is summarized in Algorithm 2.1. Since $\mathbf{\Sigma}$ is known, it is called the model-based GLR test in this study (or LR test in statistics).

In practical applications, the covariance matrix $\mathbf{\Sigma}$ is generally unknown, which needs to be identified from the data. This brings about the data-driven GLR test. It is straightforward that

$$\lim_{N\to\infty} S_{N-1} = \lim_{N\to\infty} \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{z}_k - \bar{\mathbf{z}})(\mathbf{z}_k - \bar{\mathbf{z}})^T = E((\mathbf{z} - E(\mathbf{z}))(\mathbf{z} - E(\mathbf{z}))^T) = \mathbf{\Sigma}$$

which delivers an unbiased estimate of the covariance matrix. Thus if there are sufficient training data available (a large enough N), the unknown parameter $\mathbf{\Sigma}$ could be approximated using the above equation. The next steps for carrying out the GLR test are the same as the model-based version. However, for non-sufficient test data (N is not large enough), the test statistic J_{GLR} will deviate from the chi-squared distribution. The consequence is that the resulting threshold will not match the specified significance level, which will finally decrease the monitoring performance.

The Hotelling's T^2 statistic [58, 80] provides one solution to the aforementioned problem. Its derivation will be addressed in the following.

Suppose two data sets $\mathbf{z}_i, i = 1, \cdots, N_0$, and $\mathbf{z}_i, i = 1, \cdots, N_1$, with the same covariance matrix $\mathbf{\Sigma}$ are available, then

$$\mathbf{S}_0 = \frac{1}{N_0 - 1}\sum_{k=1}^{N_0}(\mathbf{z}_k - \bar{\mathbf{z}}_0)(\mathbf{z}_k - \bar{\mathbf{z}}_0)^T,\ \bar{\mathbf{z}}_0 = \frac{1}{N_0}\sum_{i=1}^{N_0}\mathbf{z}_k, \tag{2.12}$$

$$\mathbf{S}_1 = \frac{1}{N_1 - 1}\sum_{k=1}^{N_1}(\mathbf{z}_k - \bar{\mathbf{z}}_1)(\mathbf{z}_k - \bar{\mathbf{z}}_1)^T,\ \bar{\mathbf{z}}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{z}_k \tag{2.13}$$

provide unbiased estimates of $\mathbf{\Sigma}$, respectively. It is easy to prove that

$$\mathbf{S} = \frac{1}{N_0 + N_1 - 2}\left(\sum_{k=1}^{N_0}(\mathbf{z}_k - \bar{\mathbf{z}}_0)(\mathbf{z}_k - \bar{\mathbf{z}}_0)^T + \sum_{k=1}^{N_1}(\mathbf{z}_k - \bar{\mathbf{z}}_1)(\mathbf{z}_k - \bar{\mathbf{z}}_1)^T\right)$$

is an unbiased estimate of $\mathbf{\Sigma}$ as well, *i.e.* $E(\mathbf{S}) = \mathbf{\Sigma}$.

Since $(N_0 + N_1 - 2)\mathbf{S}$ follows the Wishart distribution [58, 80]

$$(N_0 + N_1 - 2)\mathbf{S} \sim \mathcal{W}_m(N_0 + N_1 - 2, \mathbf{\Sigma})$$

and

$$\sqrt{\frac{N_0 N_1}{N_0 + N_1}}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0) \sim \mathcal{N}(0, \mathbf{\Sigma}),$$

Algorithm 2.2. *Hotelling's T^2 test*

Based on the offline data $\mathbf{z}_k,\ k = 1, \cdots, N_0$,

S1: Compute the estimates $\bar{\mathbf{z}}_0$ *and* $\mathbf{S}$ *as (2.12).*

S2: Determine $\mathcal{F}_{1-\alpha}(n_\mathbf{z}, N_0 - n_\mathbf{z})$ *using the* $\mathcal{F}$*-distribution with* $(n_\mathbf{z}, N_0 - n_\mathbf{z})$ *degrees of freedom for a significance level* α.

S3: Set the threshold: $J_{T^2_{th}} = \frac{n_\mathbf{z}(N_0^2-1)}{N_0(N_0-n_\mathbf{z})}\mathcal{F}_{1-\alpha}(n_\mathbf{z}, N_0 - n_\mathbf{z})$.

Based on the online sample $\mathbf{z}_k$,

S4: Construct the test statistic: $T^2 = (\mathbf{z}_k - \bar{\mathbf{z}}_0)^T\mathbf{S}_0^{-1}(\mathbf{z}_k - \bar{\mathbf{z}}_0)$.

S5: Check the decision logic:

$$\begin{cases} T^2 > J_{T^2_{th}} \Rightarrow & H_1 : \textit{faulty} \\ T^2 \leq J_{T^2_{th}} \Rightarrow & H_0 : \textit{fault-free.} \end{cases}$$

we have

$$\gamma := \frac{N_0N_1}{N_0 + N_1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)^T\mathbf{\Sigma}^{-1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0) \sim \chi^2(n_\mathbf{z}), \tag{2.14}$$

$$\beta := \frac{\frac{N_0N_1}{N_0+N_1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)^T\mathbf{\Sigma}^{-1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)}{\frac{N_0N_1}{N_0+N_1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)^T((N_0 + N_1 - 2)\mathbf{S})^{-1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)} \sim \chi^2((N_0 + N_1 - 2) - n_\mathbf{z} + 1). \tag{2.15}$$

It follows that

$$\frac{\gamma/n_\mathbf{z}}{\beta/(N_0 + N_1 - n_\mathbf{z} - 1)} \sim \mathcal{F}(n_\mathbf{z}, N_0 + N_1 - n_\mathbf{z} - 1). \tag{2.16}$$

By substituting (2.14) and (2.15) into (2.16), it gives

$$\begin{aligned} &\frac{N_0N_1(N_0 + N_1 - n_\mathbf{z} - 1)}{n_\mathbf{z}(N_0 + N_1 - 2)(N_0 + N_1)}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)^T\mathbf{S}^{-1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0) \sim \mathcal{F}(n_\mathbf{z}, N_0 + N_1 - n_\mathbf{z} - 1) \\ \iff &(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)^T\mathbf{S}^{-1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0) \sim \frac{n_\mathbf{z}(N_0 + N_1 - 2)(N_0 + N_1)}{N_0N_1(N_0 + N_1 - n_\mathbf{z} - 1)}\mathcal{F}(n_\mathbf{z}, N_0 + N_1 - n_\mathbf{z} - 1). \end{aligned}$$

The Hotelling's T^2 statistic is defined as

$$T^2_{GLR} = (\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)^T\mathbf{S}^{-1}(\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_0)$$

where $\mathbf{S}$ is an unbiased estimate of the covariance matrix.

The corresponding threshold can be determined by the $\mathcal{F}$-distribution as

$$J_{T^2_{GLR,th}} = \frac{n_\mathbf{z}(N_0 + N_1 - 2)(N_0 + N_1)}{N_0N_1(N_0 + N_1 - n_\mathbf{z} - 1)}\mathcal{F}_{1-\alpha}(n_\mathbf{z}, N_0 + N_1 - n_\mathbf{z} - 1)$$

where α is a user defined significance level.

Note that in the case of $N_1 = 1$,

$$\mathbf{S} = \frac{1}{N_0 + 1 - 2}\left(\sum_{k=1}^{N_0}(\mathbf{z}_k - \bar{\mathbf{z}}_0)(\mathbf{z}_k - \bar{\mathbf{z}}_0)^T + 0\right) = \mathbf{S}_0,$$

i.e. only the offline data is utilized. The test statistic for a single online sample and the threshold are

$$T^2 = (\mathbf{z}_k - \bar{\mathbf{z}}_0)^T \mathbf{S}_0^{-1}(\mathbf{z}_k - \bar{\mathbf{z}}_0),$$
$$J_{T^2_{th}} = \frac{n_{\mathbf{z}}(N_0^2 - 1)}{N_0(N_0 - n_{\mathbf{z}})}\mathcal{F}_{1-\alpha}(n_{\mathbf{z}}, N_0 - n_{\mathbf{z}}).$$

The Hotelling's T^2 test procedures (for single online sample) are summarized in Algorithm 2.2.

2.2.2 Principal component analysis based technique

In modern automation processes, a great amount of data are available which contain strongly redundant information. This fact makes the direct application of the GLR-based algorithms impractical due to numerical reasons, *e.g.* the reversibility of covariance matrix or its estimate. As a result, the principal component analysis (PCA) tool [91], which is originally developed for dimension reduction, is applied to the process monitoring. Application of PCA to the process monitoring consists of two phases, *i.e.* the offline training phase and the online monitoring phase.

Denote the offline training data as

$$\mathbf{Z}_{obs} = \begin{bmatrix}\mathbf{z}(1) & \cdots & \mathbf{z}(N)\end{bmatrix} \in \mathcal{R}^{n_{\mathbf{z}} \times N} \tag{2.17}$$

where $\mathbf{z}(i) \in \mathcal{R}^{n_{\mathbf{z}}}, i = 1, \cdots, N$, denotes a sample vector. By applying the procedure given in (2.7) and (2.8), the normalized data matrix $\mathbf{Z}$ is obtained, *i.e.*

$$\mathbf{Z} = \mathrm{diag}(\mathrm{std}(\mathbf{Z}_{obs}^T).^{-1})(\mathbf{Z}_{obs} - \mathrm{mean}(\mathbf{Z}_{obs}^T)^T \mathbf{1}_N^T])$$

where $\mathbf{1}_N \in \mathcal{R}^N$ is a column vector with all elements equal one.

PCA aims at solving the following optimization problem recursively for $i = 1, \cdots, \gamma$,

$$\mathbf{p}_i = \underset{||\mathbf{p}_i||=1,\ \mathbf{p}_i^T \mathbf{p}_j = 0\ (i \neq j)}{\arg\max}\ \mathbf{p}_i^{*T} \frac{\mathbf{Z}_i \mathbf{Z}_i^T}{N-1} \mathbf{p}_i^*$$

Algorithm 2.3. *PCA-based process monitoring*

Based on the normalized offline data $\mathbf{Z} \in \mathcal{R}^{n_\mathbf{z} \times N}$,

S1: *Compute* $\mathbf{P}_{pc}$, $\mathbf{\Lambda}_{pc}$ *and* $\mathbf{\Lambda}_{res}$ *according to (2.18).*

S2: *Determine* $\chi^2_{1-\alpha}(\gamma)$ *using the* χ^2*-distribution with* γ *degrees of freedom for a significance level* α *and* c_α *using the normal distribution with probability* $1-\alpha$.

S3: *Set the thresholds* $J_{T^2,th}$ *and* $J_{SPE,th}$ *according to (2.21) and (2.22).*

Based on the normalized online sample $\mathbf{z}_k$,

S4: *Construct the test statistics:* $T^2 = \mathbf{z}_k^T \mathbf{P}_{pc} \mathbf{\Lambda}_{pc}^{-1} \mathbf{P}_{pc}^T \mathbf{z}_k^T$ *and* $SPE = \mathbf{z}_k^T (I - \mathbf{P}_{pc}\mathbf{P}_{pc}^T)\mathbf{z}_k$.

S5: *Check the decision logic:*

$$\begin{cases} T^2 > J_{T^2_{th}} \text{ or } SPE > J_{SPE,th} \Rightarrow & H_1 : \textit{faulty} \\ T^2 \leq J_{T^2_{th}} \text{ and } SPE \leq J_{SPE,th} \Rightarrow & H_0 : \textit{fault-free.} \end{cases}$$

where $\mathbf{Z}_{i+1} = (\mathbf{I} - \mathbf{p}_i \mathbf{p}_i^T)\mathbf{Z}_i, i = 1, \cdots, \gamma - 1, \mathbf{Z}_1 = \mathbf{Z}$, denotes the deflected data matrix and γ is the number of principal components. PCA extracts those γ orthogonal directions which contain the most significant variability/useful information. The recursive optimization problem can be solved by doing an eigenvalue decomposition (EVD) or SVD as

$$\begin{bmatrix} \mathbf{P}_{pc} & \mathbf{P}_{res} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_{pc} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{res} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{pc}^T \\ \mathbf{P}_{res}^T \end{bmatrix} = \frac{\mathbf{Z}\mathbf{Z}^T}{N-1} \tag{2.18}$$

where $\mathbf{P}_{pc} = [\mathbf{p}_1, \cdots, \mathbf{p}_\gamma] \in \mathcal{R}^{n_\mathbf{z} \times \gamma}$ and $\mathbf{P}_{res} = [\mathbf{p}_{\gamma+1}, \cdots, \mathbf{p}_{n_\mathbf{z}}] \in \mathcal{R}^{n_\mathbf{z} \times (n_\mathbf{z} - \gamma)}$ contain the loading vectors for the principal components and the residual components, respectively; $\mathbf{\Lambda}_{pc} = \text{diag}(\lambda_1, \cdots, \lambda_\gamma)$ and $\mathbf{\Lambda}_{res} = \text{diag}(\lambda_{\gamma+1}, \cdots, \lambda_{n_\mathbf{z}})$ are the scores of them with $\lambda_1 \geq \cdots \geq \lambda_\gamma >> \lambda_{\gamma+1} \geq \cdots \geq \lambda_{n_\mathbf{z}}$.

For detecting process abnormalities, the Hotelling's T^2 and the squared prediction error (SPE) test statistics are defined as

$$T^2 = \mathbf{z}^T \mathbf{P}_{pc} \mathbf{\Lambda}_{pc}^{-1} \mathbf{P}_{pc}^T \mathbf{z}, \tag{2.19}$$

$$SPE = \mathbf{z}^T (\mathbf{I} - \mathbf{P}_{pc}\mathbf{P}_{pc}^T)\mathbf{z}. \tag{2.20}$$

The corresponding thresholds are given as

$$J_{T^2,th} = \frac{\gamma(N^2-1)}{N(N-\gamma)} \mathcal{F}_{1-\alpha}(\gamma, N-\gamma), \tag{2.21}$$

$$J_{SPE,th} = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0}, \tag{2.22}$$

$$\theta_i = \sum_{j=\gamma+1}^{n_\mathbf{z}} \lambda_j^i, \; i = 1, 2, 3, \; h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \tag{2.23}$$

where α is the significance level and c_α the normal deviate corresponding to the upper $1-\alpha$ percentile. The statistical properties of the SPE statistic and its threshold are given in [54]. The test procedures of PCA-based process monitoring is summarized in Algorithm 2.3.

2.2.3 Partial least squares regression based technique

Partial least squares (PLS) regression is another multivariate statistical tool applied for processing redundant information in huge amount of process data. It was originally developed by H. Wold [115] for solving the co-linearity problem of least squares regression. The pioneer work on the application of PLS for process monitoring includes the one reported in [78], which aims at monitoring the quality-related low-level process variable subspace.

Denote the normalized data matrices for the low-level process variables and the high-level KPIs as

$$\mathbf{Y} = [\mathbf{y}(1), \cdots, \mathbf{y}(N)] \in \mathcal{R}^{m\times N} \text{ and } \mathbf{\Theta} = [\boldsymbol{\theta}(1), \cdots, \boldsymbol{\theta}(N)] \in \mathcal{R}^{l\times N},$$

the objective of PLS is achieved by solving the following optimization problem recursively for $i = 1, \cdots, \gamma$

$$(\mathbf{w}_i^*, \mathbf{v}_i^*) = \underset{||\mathbf{w}_i||=1, ||\mathbf{v}_i||=1}{\arg\max} \ \mathbf{w}_i^T \mathbf{Y}_i \mathbf{\Theta}^T \mathbf{v}_i \tag{2.24}$$

$$\mathbf{t}_i = \mathbf{Y}_i^T \mathbf{w}_i^*, \ \mathbf{p}_i = \frac{\mathbf{Y}_i \mathbf{t}_i}{||\mathbf{t}_i||^2}, \ \mathbf{q}_i = \frac{\mathbf{\Theta}\mathbf{t}_i}{||\mathbf{t}_i||^2} \tag{2.25}$$

$$\mathbf{r}_i = \begin{cases} \mathbf{w}_1^*, & i = 1 \\ \prod_{k=1}^{i-1} (\mathbf{I} - \mathbf{w}_k^* \mathbf{p}_k^T) \mathbf{w}_i^*, & i > 1 \end{cases} \tag{2.26}$$

where $\mathbf{Y}_{i+1} = \mathbf{Y}_i(\mathbf{I} - \frac{\mathbf{t}_i \mathbf{t}_i^T}{||\mathbf{t}_i||^2})$, $\mathbf{Y}_1 = \mathbf{Y}$, and γ is the pre-determined latent variable number. Note that for each i, $\mathbf{w}_i^*$ and $\mathbf{v}_i^*$ are selected in such a way that the covariance between $\mathbf{w}_i^{*T}\mathbf{Y}_i$ and $\mathbf{v}_i^{*T}\mathbf{\Theta}$ is maximized. By iteratively solving the above problem, the following parameter matrices can be obtained:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \cdots & \mathbf{r}_\gamma \end{bmatrix}, \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_\gamma \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_\gamma \end{bmatrix}, \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_\gamma^T \end{bmatrix} = \mathbf{R}^T\mathbf{Y}$$

where $\mathbf{T}$ denotes data matrix of the latent variables. In addition, the estimation of process variables and KPIs based on the latent variables can be established as

$$\hat{\mathbf{Y}}_\gamma = \mathbf{P}\mathbf{R}^T\mathbf{Y}, \ \hat{\mathbf{\Theta}}_\gamma = \mathbf{Q}\mathbf{R}^T\mathbf{Y}.$$

Algorithm 2.4. *PLS-based process monitoring*

Based on the normalized data matrices $\mathbf{Y}$ *and* $\mathbf{\Theta}$*,*

S1: Set $\mathbf{Y}_1 = \mathbf{Y}$ *and, for* $i = 1, \cdots, \gamma$*, recursively compute*

$$(\mathbf{w}_i^*, \sigma_{max}, \mathbf{v}_i^*) = \mathrm{svd}(\mathbf{Y}_i \mathbf{\Theta}^T),$$

$$\mathbf{t}_i = \mathbf{Y}_i^T \mathbf{w}_i^*,\ p_i = \frac{\mathbf{Y}_i \mathbf{t}_i}{||\mathbf{t}_i||^2},\ \mathbf{q}_i = \frac{\mathbf{\Theta} \mathbf{t}_i}{||\mathbf{t}_i||^2},\ \mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{p}_i \mathbf{t}_i^T$$

where σ_{max} *is the maximal singular value of* $\mathbf{Y}_i\mathbf{\Theta}$*,* $\mathbf{w}_i^*$ *and* $\mathbf{v}_i^*$ *are the corresponding singular vectors,* γ *is the pre-determined latent variable number.*

S2: Form the matrices $\mathbf{P}, \mathbf{R}$ *and* $\mathbf{\Lambda}_\gamma$*:*

$$\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_\gamma] \in \mathcal{R}^{m \times \gamma},\ \mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1} \in \mathcal{R}^{m \times \gamma},\ \mathbf{\Lambda}_\gamma = \mathrm{diag}\left(\frac{||\mathbf{t}_1||^2}{N-1}, \cdots, \frac{||\mathbf{t}_m||^2}{N-1}\right).$$

S3: Determine $\mathcal{F}_{1-\alpha}(\gamma, N-\gamma)$ *and* $\chi^2_\alpha(h)$ *using the* $\mathcal{F}$*- and* χ^2*-distributions.*

S4: Set the thresholds $J_{T^2,th}$ *and* $J_{SPE,th}$ *according to (2.31) and (2.32).*

Based on the normalized online sample $\mathbf{y}_k$*,*

S5: Construct the test statistics: $T^2 = \mathbf{y}_k^T \mathbf{R} \mathbf{\Lambda}_\gamma^{-1} \mathbf{R}^T \mathbf{y}_k$ *and* $SPE = \mathbf{y}_k^T (\mathbf{I} - \mathbf{P}\mathbf{R}^T)\mathbf{y}_k$*.*

S6: Check the decision logic:

$$\begin{cases} T^2 > J_{T^2_{th}} \text{ and } SPE \le J_{SPE,th} \Rightarrow & \textit{fault influences KPI} \\ T^2 \le J_{T^2_{th}} \text{ and } SPE > J_{SPE,th} \Rightarrow & \textit{fault does not influence KPI} \\ T^2 > J_{T^2_{th}} \text{ and } SPE > J_{SPE,th} \Rightarrow & \textit{both kinds of faults happen} \\ T^2 \le J_{T^2_{th}} \text{ and } SPE \le J_{SPE,th} \Rightarrow & \textit{fault-free.} \end{cases}$$

For monitoring the latent variable space whose dimension is significantly lower than m, Theorem 2.1 plays a key role.

Theorem 2.1. *Based on the standard PLS regression method, the following relation holds:*

$$\mathbf{t}_i^T \mathbf{t}_j = 0,\ \textit{for } i \neq j. \tag{2.27}$$

The proof is given in Appendix A. Theorem 2.1 concludes that the latent variables are mutually uncorrelated. It is a very nice property for the process monitoring.

In order to monitor KPI-related low-level process variables, the following test statistics are established:

$$T^2 = \mathbf{y}^T \mathbf{R} \mathbf{\Lambda}_\gamma^{-1} \mathbf{R}^T \mathbf{y} \tag{2.28}$$

$$SPE = \mathbf{y}^T (\mathbf{I} - \mathbf{P}\mathbf{R}^T)\mathbf{y} \tag{2.29}$$

where

$$\mathbf{\Lambda}_\gamma = \mathrm{diag}\left(\frac{||\mathbf{t}_1||^2}{N-1}, \cdots, \frac{||\mathbf{t}_m||^2}{N-1}\right). \tag{2.30}$$

The corresponding thresholds are given as

$$J_{T^2,th} = \frac{\gamma(N^2-1)}{N(N-\gamma)}\mathcal{F}_{1-\alpha}(\gamma, N-\gamma), \tag{2.31}$$

$$J_{SPE,th} = g\chi^2_{1-\alpha}(h) \tag{2.32}$$

where α is the significance level, $g = \frac{\tilde{S}}{2\mu_{SPE}}$ is a scaling factor and $h = \frac{2\mu^2_{SPE}}{\tilde{S}}$ is the degrees of freedom of the χ^2-distribution with μ_{SPE} and $\tilde{S}$ denoting the sample mean and covariance of SPE. The complete procedures for PLS-based process monitoring is summarized in Algorithm 2.4.

2.3 Monitoring techniques for lumped-parameter processes

Monitoring techniques for the lumped-parameter processes, whose dynamics is governed by the following ODEs, *i.e.*

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}\mathbf{y}(k) + \boldsymbol{\eta}(k) + \mathbf{E}\mathbf{f}(k),\ \mathbf{z}(0) = \mathbf{z}_0, \tag{2.33}$$

$$\boldsymbol{\theta}(k) = \mathbf{C}\mathbf{z}(k) + \mathbf{D}\mathbf{y}(k) + \boldsymbol{\xi}(k) + \mathbf{F}\mathbf{f}(k) \tag{2.34}$$

as given in (2.4-2.5), have received rapid development during the past 50 years and are well established. The representational techniques include the FDF, the DO scheme and the PS approach.

2.3.1 Fault detection filter based residual generation

Considering the process described by (2.33-2.34), in order to monitor its operation condition, a full order state observer [24] could be applied, *i.e.*

$$\begin{aligned}\hat{\mathbf{z}}(k+1) &= \mathbf{A}\hat{\mathbf{z}}(k) + \mathbf{B}\mathbf{y}(k) + \mathbf{L}(\boldsymbol{\theta}(k) - \mathbf{C}\hat{\mathbf{z}}(k) - \mathbf{D}\mathbf{y}(k))\\ \mathbf{r}(k) &= \mathbf{V}(\boldsymbol{\theta}(k) - \mathbf{C}\hat{\mathbf{z}}(k) - \mathbf{D}\mathbf{y}(k))\end{aligned}$$

Denote $\mathbf{e}(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k)$, we have the dynamics of the FDF [24]

$$\begin{aligned}\mathbf{e}(k+1) &= (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}(k) + \boldsymbol{\eta}(k) - \mathbf{L}\boldsymbol{\xi}(k) + (\mathbf{E} - \mathbf{L}\mathbf{F})\mathbf{f}(k)\\ \mathbf{r}(k) &= \mathbf{V}\mathbf{C}\mathbf{e}(k) + \mathbf{V}\boldsymbol{\xi}(k) + \mathbf{V}\mathbf{F}\mathbf{f}(k)\end{aligned}$$

where $\mathbf{L}$ and $\mathbf{V}$ are the free design parameters. $\mathbf{L}$ should be selected in such a way that $\mathbf{A}-\mathbf{L}\mathbf{C}$ is stable, *i.e.* the eigenvalues are inside the unit circle. Note that in the fault- and

disturbance-free case, $\lim_{k\to\infty} \mathbf{e}(k) = 0 \Rightarrow \lim_{k\to\infty} \mathbf{r}(k) = 0$, which indicates the healthy/normal condition of the process. If a fault happens, $\lim_{k\to\infty} \mathbf{r}(k) \neq 0$ indicates the occurrence of the fault. In industrial automation processes, however, disturbances are inevitable, $\mathbf{r}(k) \neq 0$ cannot be used to make any decision. Thus the residual signal $\mathbf{r}(k)$ should be evaluated and compared with some threshold, by using its statistical information.

2.3.2 Diagnostic observer based residual generation

Different from the FDF, DO is an output observer based approach [24] described by

$$\begin{aligned} \mathbf{z}_d(k+1) &= \mathbf{G}\mathbf{z}_d(k) + \mathbf{H}\mathbf{y}(k) + \mathbf{L}\boldsymbol{\theta}(k) \\ r(k) &= \mathbf{v}\boldsymbol{\theta}(k) - \mathbf{w}\mathbf{z}_d(k) - \mathbf{q}\mathbf{y}(k) \end{aligned}$$

where $\mathbf{z}_d(k) = \mathbf{T}\mathbf{z}(k) \in \mathcal{R}^s$ denotes the state variables. The order of the observer s could be different from the system order n. The parameters as well as $\mathbf{T}$ have to satisfy the following conditions:

- $\mathbf{G}$ is stable,
- $\mathbf{TA} - \mathbf{GT} = \mathbf{LC}$, $\mathbf{H} = \mathbf{TB} - \mathbf{LD}$, and
- $\mathbf{vC} - \mathbf{wT} = \mathbf{0}$, $\mathbf{q} = \mathbf{vD}$.

Denote $\mathbf{e}(k) = \mathbf{T}\mathbf{z}(k) - \mathbf{z}_d(k)$, the dynamics of the DO is governed by

$$\begin{aligned} \mathbf{e}(k+1) &= \mathbf{G}\mathbf{e}(k) + \mathbf{T}\boldsymbol{\eta}(k) - \mathbf{L}\boldsymbol{\xi}(k) + (\mathbf{TE} - \mathbf{LF})\mathbf{f}(k) \\ r(k) &= \mathbf{we}(k) + \mathbf{v}\boldsymbol{\xi}(k) + \mathbf{vF}\mathbf{f}(k) \end{aligned}$$

Under the aforementioned three conditions, it is obvious that in the disturbance- and fault-free case, the residual signal $r(k)$ equals zero in the steady state. Any disturbance or fault could make $r(k)$ diverge from zero. Thus in practice a threshold should be set based on the statistical property of the disturbances.

2.3.3 Parity space based residual generation

PS-based residual generation [24] belongs to the most straightforward fault detection methods which has a very clear geometrical meaning. It is assumed that there is no redundancy in the KPIs, *i.e.* $rank(\mathbf{C}) = l$. Given the PS order s, the process (2.33-2.34) could be extended as

$$\boldsymbol{\theta}_s(k) = \boldsymbol{\Gamma}_s\mathbf{z}(k-s) + \mathbf{H}_{y,s}\mathbf{y}_s(k) + \mathbf{H}_{\boldsymbol{\eta},s}\boldsymbol{\eta}_s(k) + \mathbf{H}_{\boldsymbol{\xi},s}\boldsymbol{\xi}_s(k) + \mathbf{H}_{f,s}\mathbf{f}_s(k)$$

where $\boldsymbol{\theta}_s(k), \mathbf{y}_s(k), \boldsymbol{\eta}_s(k), \boldsymbol{\xi}_s(k)$ and $\mathbf{f}_s(k)$ are constructed as $\boldsymbol{\pi}_s(k)$ with the following data structure

$$\boldsymbol{\pi}_s(k) = \begin{bmatrix} \boldsymbol{\pi}(k-s) \\ \boldsymbol{\pi}(k-s+1) \\ \cdots \\ \boldsymbol{\pi}(k) \end{bmatrix} \tag{2.35}$$

and

$$\boldsymbol{\Gamma}_s = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^s \end{bmatrix}, \mathbf{H}_{y,s} = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{CA}^{s-1}\mathbf{B} & \cdots & \mathbf{CB} & \mathbf{D} \end{bmatrix}, \mathbf{H}_{f,s} = \begin{bmatrix} \mathbf{F} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CE} & \mathbf{F} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{CA}^{s-1}\mathbf{E} & \cdots & \mathbf{CE} & \mathbf{F} \end{bmatrix}, \tag{2.36}$$

$$\mathbf{H}_{\boldsymbol{\eta},s} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{CA}^{s-1} & \cdots & \mathbf{C} & \mathbf{0} \end{bmatrix}, \mathbf{H}_{\boldsymbol{\xi},s} = \mathbf{I} \in \mathcal{R}^{(s+1)l \times (s+1)l}. \tag{2.37}$$

The PS-based residual generator can be constructed as

$$r_s(k) = \mathbf{v}_s \left(\boldsymbol{\theta}_s(k) - \mathbf{H}_{y,s}\mathbf{y}_s(k) - \boldsymbol{\Gamma}_s\mathbf{z}(k-s) - \mathbf{H}_{\boldsymbol{\eta},s}\boldsymbol{\eta}_s(k) - \mathbf{H}_{\boldsymbol{\xi},s}\boldsymbol{\xi}_s(k) - \mathbf{H}_{f,s}\mathbf{f}_s(k) \right). \tag{2.38}$$

Note that in the disturbance- and fault-free case, if $v_s \in \mathcal{R}^{1\times(s+1)l}$ is selected from the orthogonal subspace of the subspace spanned by the columns of $\boldsymbol{\Gamma}_s$, then we have $r_s(k) = \mathbf{0}$. The parity vector $\mathbf{v}_s$ is expected to be determined in such a way that the disturbances are completely decoupled while the fault is completely reflected by $r_s(k)$. In practice, if a perfect decoupling of the disturbances is infeasible, then the influence of the disturbances on $r_s(k)$ should be minimized.

2.3.4 PS-based design and DO-based implementation

From the previous subsections, it is clear that the requirements of the PS-based method on computation cost and memory storage are higher than the DO-based approach while the design effort is much lower. It has been proven in [24] that the two approaches are equivalent, which makes the popular "PS-based design and DO-based implementation scheme" feasible. Algorithm 2.5 provides the basic procedures of this scheme.

Algorithm 2.5. *PS-based design and DO-based implementation scheme*

Given the process model (2.4-2.5),

S1: *Build* $\mathbf{\Gamma}_s$ *and select a parity vector* $\mathbf{v}_s := [\mathbf{v}_{s,0}, \mathbf{v}_{s,1}, \cdots, \mathbf{v}_{s,s-1}, \mathbf{v}_{s,s}] \in \mathcal{R}^{(s+1)l}$ *from the orthogonal subspace of the column subspace of* $\mathbf{\Gamma}_s$.

S2: *Construct* $\mathbf{T}$ *as*

$$\mathbf{T} = \begin{bmatrix} \mathbf{v}_{s,1} & \cdots & \mathbf{v}_{s,s-1} & \mathbf{v}_{s,s} \\ \mathbf{v}_{s,2} & \cdots & \mathbf{v}_{s,s} & \mathbf{0} \\ \vdots & \iddots & \iddots & \vdots \\ \mathbf{v}_{s,s} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \vdots \\ \mathbf{C}\mathbf{A}^{s-2} \\ \mathbf{C}\mathbf{A}^{s-1} \end{bmatrix}.$$

S3: *Set*

$$\mathbf{G} = [\mathbf{G}_0, \mathbf{g}], \mathbf{G}_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathcal{R}^{s\times(s-1)}, \ \mathbf{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_s \end{bmatrix} \in \mathcal{R}^s$$

which should be stable.

S4: *Construct*

$$\mathbf{H} = \begin{bmatrix} \mathbf{v}_{s,0} + g_1\mathbf{v}_{s,s} & \mathbf{v}_{s,1} & \cdots & \cdots & \mathbf{v}_{s,s} \\ \mathbf{v}_{s,1} + g_2\mathbf{v}_{s,s} & \mathbf{v}_{s,2} & \cdots & \mathbf{v}_{s,s} & \mathbf{0} \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ \mathbf{v}_{s,s-1} + g_s\mathbf{v}_{s,s} & \mathbf{v}_{s,s} & \iddots & \iddots & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ \mathbf{C}\mathbf{B} \\ \vdots \\ \mathbf{C}\mathbf{A}^{s-1}\mathbf{B} \end{bmatrix}, \ \mathbf{L} = -\begin{bmatrix} \mathbf{v}_{s,0} \\ \mathbf{v}_{s,1} \\ \vdots \\ \mathbf{v}_{s,s-1} \end{bmatrix} - \mathbf{g}\mathbf{v}_{s,s},$$

$$\mathbf{v} = \mathbf{v}_{s,s}, \ \mathbf{w} = [0, \cdots, 0, 1], \ \mathbf{q} = \mathbf{v}_{s,s}\mathbf{D}.$$

2.4 Monitoring techniques for distributed-parameter processes

There are quite a few approaches for monitoring the DPPs. Consider a class of system represented in an abstract space

$$\dot{z}(t) = \mathcal{A}z(t) + \mathcal{B}y(t)$$

where $\mathcal{A}$ is the system operator and $\mathcal{B}$ is the input operator [21]. To reduce the dimension, the infinite dimensional state is decomposed as

$$z(t) = z_s(t) + z_f(t)$$

where z_s represents the state of the finite dimensional slow subsystem and z_f denotes the state of the infinite dimensional fast and stable system. Using it, the original description

can be written as

$$\dot{z}_s(t) = \mathcal{A}_s z_s(t) + \mathcal{B}_s y(t),\ \dot{z}_f(t) = \mathcal{A}_f z_f(t) + \mathcal{B}_f y(t).$$

By neglecting the fast subsystem, a detection observer is designed for the slow subsystem. The state prediction error is considered as the residual signal. The random disturbances/noise are/is not considered and the monitoring system is established in the deterministic framework. Moreover, the threshold is determined analytically using the knowledge of the deterministic disturbances.

Remarks: A very strong assumption for this approach is the existence of the eigen-decomposition of the operator $\mathcal{A}$, which is only well-known for some systems [19]. In the general case, derivation of the eigen-decomposition of an operator is a manual task and requires great design effort (needs advanced mathematical knowledge). Similar process monitoring results based on the slow subsystem include [4, 32].

2.5 Concluding remarks

This chapter summarizes the basics for modern process monitoring system design. Depending on the significance of the process dynamics and the application scope, industrial automation processes can be modelled as static processes, LPPs and DPPs. The multivariate statistical techniques are powerful tools for monitoring the large-scale processes. They are generally applicable for static processes and do not require any first-principle model. The existing methods are originally developed in the mathematical domain. They are very complex and not optimal for the process monitoring. Thus the forthcoming chapter discusses an alternative monitoring approach which provides optimal monitoring performance.

Another drawback of the traditional multivariate statistical approaches lies in dealing with process dynamics. Although some dynamical modifications have been made in the research area, their performance is quite limited. On the other hand, the model-based process monitoring techniques are efficient in handling dynamic issues. The techniques have been well established, but they put very high requirements on accurate first-principle models. Thus it is appealing to have data-driven design schemes for model-based techniques. This issue is further addressed in Chapter 4.

Performance monitoring of DPPs is now basically a blank field in the research. Available techniques have too strong assumptions on the types of DPPs (*e.g.* the eigen-decomposition is known and fulfills certain condition) and they do not consider stochastic factors. Chapter 5 aims at establishing a stochastic monitoring framework for DPPs without those restrictions. In addition, its data-driven realization is another import goal.

3 An alternative data-driven KPI monitoring scheme for static processes

For many automation processes, the operating conditions are kept unchanged due to mass production and the processes are in steady state for most of the time. Thus static key performance monitoring schemes are widely implemented due to their excellent applicability. Among various static monitoring schemes, PLS-based technique is the most popular. PLS, originally called NIPALS (nonlinear iterative partial least squares), was first developed by H. Wold [115] in the 1960s in the economic area. It is an effective modelling approach of general scope for cause-effect inference and prediction where the concept of *latent variable* plays a key role in handling collinearity among independent variables [115]. With rapid application of SCADA systems and digital computers, PLS has been applied in many automation processes [63]. The algorithm given in Chapter 2 is a standard one and widely used in practice. However, the monitoring performance is not optimal since the T^2 statistic contains useless information that is unrelated to the KPIs and the SPE statistic may exhibit large variations where a more accurate statistical distribution can be obtained. In addition, *cross validation*, which is a widely-used effective approach to select the latent variable number, is computationally expensive and requires great engineering effort. Motivated by these observations, a revised PLS-based scheme is first proposed for improving the monitoring performance. By analysing the PLS modelling for process monitoring purpose, an alternative scheme is then proposed to reduce the design effort while preserving high monitoring performance.

3.1 Preliminaries and problem formulation

Although not always mentioned, the data-driven methods implicitly follow some (statistical) models. For monitoring static processes, the model given in (2.1) is generally utilized. As discussed in Chapter 1, statistical redundancy is the core of a residual generator. Here the redundancy consists of two parts:

- Mean vectors $E(\mathbf{y}_{obs}), E(\boldsymbol{\theta}_{obs})$ and standard deviation vector $\boldsymbol{\lambda}_{y,std}, \boldsymbol{\lambda}_{\boldsymbol{\theta},std}$ of the process variables and the KPI data

$$E(\mathbf{y}_{obs}) \approx \frac{1}{N}\sum_{k=1}^{N}\mathbf{Y}_{obs}(:,k),\ \boldsymbol{\lambda}_{y,std} \approx \operatorname{diag}\left(\frac{(\mathbf{Y}_{obs}-E(\mathbf{y}_{obs})\mathbf{1}_N)(\mathbf{Y}_{obs}-E(\mathbf{y}_{obs})1_N)^T}{N-1}\right)$$

$$E(\boldsymbol{\theta}_{obs}) \approx \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\Theta}_{obs}(:,k),\ \boldsymbol{\lambda}_{\boldsymbol{\theta},std} \approx \operatorname{diag}\left(\frac{(\boldsymbol{\Theta}_{obs}-E(\boldsymbol{\theta}_{obs})\mathbf{1}_N)(\boldsymbol{\Theta}_{obs}-E(\boldsymbol{\theta}_{obs})1_N)^T}{N-1}\right).$$

- Parameter matrix $\boldsymbol{\Psi}$ of the following normalized process description

$$\boldsymbol{\theta} = \boldsymbol{\Psi}\mathbf{y} + \mathbf{e}_{\boldsymbol{\theta}},\ E(\mathbf{y}\mathbf{e}_{\boldsymbol{\theta}}^T) = 0. \tag{3.1}$$

In modern automation processes, a great number of process variables and KPIs are recorded in the SCADA system. In many cases, KPIs are not online measurable and the dimension of the process variables $\mathbf{y} \in \mathcal{R}^m$ is much higher than the dimension of the KPIs $\boldsymbol{\theta} \in \mathcal{R}^l$, *i.e.* $m >> l$. In addition, lots of redundant information exits in the process variable data which causes strong collinearity. It is practically impossible to apply the univariate-based techniques to monitor the process especially to identify if the faults are KPI-related or not. The objective of this chapter is to propose practical KPI monitoring schemes with the following requirements:

- The monitoring performance is optimized, *i.e.* the KPI-related test statistic contains no useless information for the KPIs and more accurate threshold is obtained for the KPI-unrelated test statistic.
- The design effort is low.
- The amount of test statistics is small.

In the following, we will first revise the standard PLS-based monitoring scheme to partially fulfil the requirements.

3.2 A revised PLS-based monitoring scheme

By running the standard PLS algorithm in Chapter 2, the latent variables are generated as

$$\mathbf{t} = \mathbf{R}^T\mathbf{y}.$$

Based on it, the estimate of the KPIs is achieved as

$$\hat{\boldsymbol{\theta}}_\gamma = \mathbf{Q}\mathbf{R}^T\mathbf{y} = (\mathbf{Q}\boldsymbol{\Lambda}_\gamma^{1/2})(\boldsymbol{\Lambda}_\gamma^{-1/2}\mathbf{R}^T\mathbf{y})$$

Algorithm 3.1. *A revised PLS-based process monitoring scheme*

S1: *Run the standard PLS algorithm to get* $\mathbf{R}$.

S2: *Compute* $\mathbf{T} = \mathbf{R}^T\mathbf{Y}, \mathbf{\Lambda}_\gamma = \mathbf{T}\mathbf{T}^T/(N-1), \mathbf{Q} = \mathbf{\Theta}T^T\mathbf{\Lambda}_\gamma^{-1}/(N-1)$.

S3: *Do an SVD on* $\mathbf{Q}\mathbf{\Lambda}_\gamma^{(1/2)}$ *as (3.2), compute* $\mathbf{R}^\perp$ *and do another SVD as (3.4).*

S4: *Determine the thresholds as:*

$$J_{th,T^2_{PLS,R}} = \chi^2_{1-\alpha}(l),\ J_{th,SPE_{PLS,R}} = d^2_{\gamma,m-l}\chi^2_{1-\alpha}(m-l).$$

Based on the normalized online process data $\mathbf{y}(k)$,

S5: *Build the test statistics* $T^2_{PLS,R}$ *and* $SPE_{PLS,R}$ *according to (3.3) and (3.5).*

S6: *Check the decision logic:*

$$\begin{cases} T^2_{PLS,R} > J_{th,T^2_{PLS,R}} and SPE_{PLS,R} \le J_{th,SPE_{PLS,R}} \Rightarrow & fault influences KPI \\ T^2_{PLS,R} \le J_{th,T^2_{PLS,R}} and SPE_{PLS,R} > J_{th,SPE_{PLS,R}} \Rightarrow & fault does not influence KPI \\ T^2_{PLS,R} > J_{th,T^2_{PLS,R}} \text{ and } SPE_{PLS,R} > J_{th,SPE_{PLS,R}} \Rightarrow & \text{both kinds of faults happen} \\ T^2_{PLS,R} \le J_{th,T^2_{PLS,R}} \text{ and } SPE_{PLS,R} \le J_{th,SPE_{PLS,R}} \Rightarrow & \text{fault-free.} \end{cases}$$

where $\mathbf{\Lambda}_\gamma$ denotes the covariance matrix of the latent variables and is obtained from (2.30). Do an SVD on $\mathbf{Q}\mathbf{\Lambda}_\gamma^{1/2}$ as

$$\mathbf{Q}\mathbf{\Lambda}_\gamma^{1/2} = \mathbf{Q}_{PLS}\begin{bmatrix}\mathbf{R}_{PLS} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{S}^T_{PLS,1} \\ \mathbf{S}^T_{PLS,2}\end{bmatrix}. \tag{3.2}$$

In order to monitor the KPI-related part in the process variable space, it is reasonable to establish the following test statistic:

$$T^2_{PLS,R} = \mathbf{y}^T\mathbf{R}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{S}_{PLS,1}\mathbf{S}^T_{PLS,1}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{R}^T\mathbf{y}. \tag{3.3}$$

Since we have $\mathbf{S}^T_{PLS,1}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{R}^T\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, thus the threshold can be determined by the χ^2-distribution with l degrees of freedom.

For monitoring the KPI-unrelated part, it is necessary to obtain $\mathbf{R}^\perp$, which has orthonormal rows and is orthogonal to $\mathbf{R}$. Then do the following SVD

$$\begin{bmatrix}\mathbf{R}^\perp \\ \mathbf{S}^T_{PLS,2}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{R}^T\end{bmatrix}\frac{\mathbf{Y}\mathbf{Y}^T}{N-1}\begin{bmatrix}(\mathbf{R}^\perp)^T & \mathbf{R}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{S}_{PLS,2}\end{bmatrix} = \mathbf{U}_\gamma\mathbf{D}_\gamma\mathbf{U}_\gamma^T \tag{3.4}$$

where $\mathbf{D}_\gamma = \mathrm{diag}(d^2_{\gamma,1}, \cdots, d^2_{\gamma,m-l})$. It is remarkable that the last singular values may be quite small. In order to improve the numerical conditions, let us define

$$\mathbf{\Omega}_\gamma = \mathrm{diag}(\frac{d^2_{\gamma,m-l}}{d^2_{\gamma,1}}, \cdots, \frac{d^2_{\gamma,m-l}}{d^2_{\gamma,m-l-1}}, 1).$$

It is meaningful to establish the KPI-unrelated test statistic as

$$SPE_{PLS,R} = \mathbf{y}^T\begin{bmatrix}(\mathbf{R}^\perp)^T & \mathbf{R}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{S}_{PLS,2}\end{bmatrix}\mathbf{U}_\gamma\mathbf{\Omega}_\gamma\mathbf{U}_\gamma^T\begin{bmatrix}\mathbf{R}^\perp \\ \mathbf{S}^T_{PLS,2}\mathbf{\Lambda}_\gamma^{-1/2}\mathbf{R}^T\end{bmatrix}\mathbf{y}. \tag{3.5}$$

The corresponding threshold can be determined using the χ^2-distribution as

$$J_{th,SPE_{PLS,R}} = d^2_{\gamma,m-l}\chi^2(m-l).$$

In Algorithm 3.1, procedures of the revised PLS-based monitoring scheme are summarized.

Remark: The revised monitoring scheme is based on the standard PLS algorithm. From the obtained latent variable data, two test statistics are established. The $T^2_{PLS,R}$ statistic contains no useless information for the KPIs and thus is essential for the KPI-based root cause analysis. In addition, a more accurate statistical distribution for the $SPE_{PLS,R}$ test statistic is established as well. As a result, the first and third requirements from *problem formulation* have been totally fulfilled. Since the standard PLS algorithm is utilized, the engineering effort is still high mainly due to the determination of the latent variable number. A numerical example is given at the end of this chapter to show these improvements. In the following section, by analysing the PLS algorithm, we will propose an alternative KPI monitoring scheme satisfying all the requirements.

3.3 An alternative KPI monitoring scheme

For static processes, the first part of the redundancy, *i.e.* the mean vectors and the standard deviation matrices, can be easily identified from the raw data easily. The challenge is how to construct the regression matrix in the normalized process description (3.1). One standard criteria for it is to minimize the total variance of prediction error $\mathbf{e}_{\boldsymbol{\theta}}$, *i.e.*

$$\text{find } \hat{\boldsymbol{\Psi}} = \arg\min \operatorname{tr}(E(\mathbf{e}_{\boldsymbol{\theta}}\mathbf{e}_{\boldsymbol{\theta}}^T)).$$

By substituting $\mathbf{e}_{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\Psi}\mathbf{y}$, we have

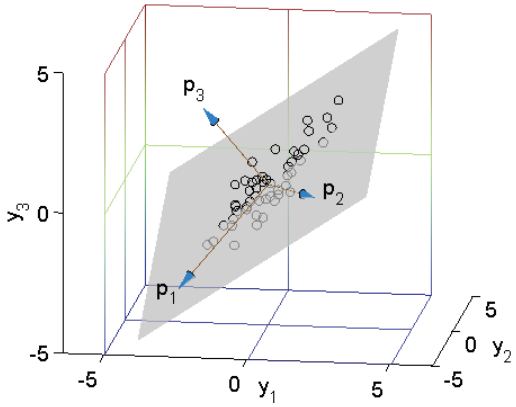
$$\operatorname{tr}(E(\mathbf{e}_{\boldsymbol{\theta}}\mathbf{e}_{\boldsymbol{\theta}}^T)) = \operatorname{tr}(E(\boldsymbol{\theta}\boldsymbol{\theta}^T)) - 2\operatorname{tr}(E(\boldsymbol{\theta}\mathbf{y}^T)\boldsymbol{\Psi}^T) + \operatorname{tr}(\boldsymbol{\Psi}E(\mathbf{y}\mathbf{y}^T)\boldsymbol{\Psi}^T).$$

Thus the minimum is achieved when $\frac{\partial \operatorname{tr}(E(\mathbf{e}_{\boldsymbol{\theta}}\mathbf{e}_{\boldsymbol{\theta}}^T))}{\partial \boldsymbol{\Psi}^T} = \mathbf{0}$, which further gives

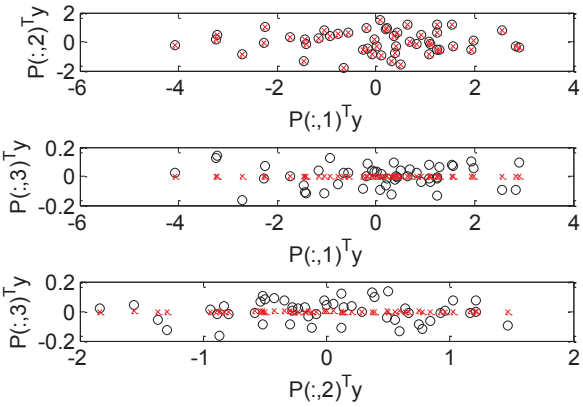
$$E(\boldsymbol{\theta}\mathbf{y}^T) - \boldsymbol{\Psi}E(\mathbf{y}\mathbf{y}^T) = \mathbf{0}.$$

Using the normalized process and KPI data $\mathbf{Y} \in \mathcal{R}^{m\times N}$ and $\boldsymbol{\Theta} \in \mathcal{R}^{l\times N}$, the above solution can be approximated as

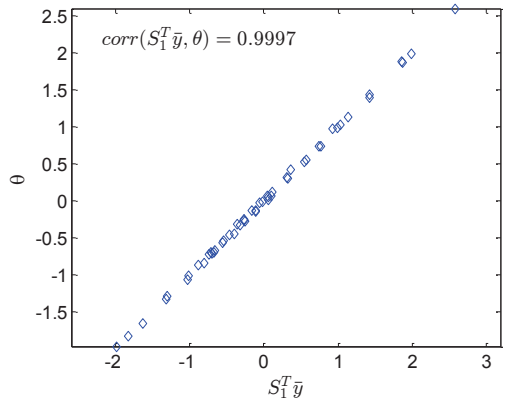
$$\frac{\boldsymbol{\Theta}\mathbf{Y}^T}{N-1} \approx \boldsymbol{\Psi}\frac{\mathbf{Y}\mathbf{Y}^T}{N-1}.$$



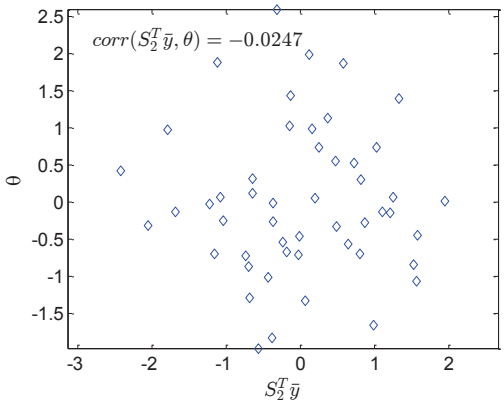
(a) Normalized data set, $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ denote directions of data variability in descending order



(b) 'o': original data; 'x': de-noised data



(c) Correlation between $\mathbf{S}_1^T\overline{\mathbf{y}}$ and KPI is strong



(d) Correlation between $\mathbf{S}_2^T\overline{\mathbf{y}}$ and KPI is weak

Figure 3.1: Illustration of the alternative algorithm

Since generally a huge number of process variables are available in reality and $\mathbf{YY}^T$ is rank deficient, $\mathbf{\Psi}$ cannot directly be identified. PLS algorithm provides one practical solution for it:

$$\hat{\mathbf{\Psi}}_{PLS} = \mathbf{QR}^T = \mathbf{\Theta Y}^T\mathbf{R}(\mathbf{R}^T\mathbf{YY}^T\mathbf{R})^{-1}\mathbf{R}^T. \tag{3.6}$$

In order to handle the collinearity problem, PLS first transforms the process variables to the low-dimensional latent variables. Due to $(\mathbf{w}_i^*, \mathbf{v}_i^*) = \underset{||\mathbf{w}_i||=1,||\mathbf{v}_i||=1}{\arg\max}\ \mathbf{w}_i^T\mathbf{Y}_i\mathbf{\Theta}^T\mathbf{v}_i$, $\mathbf{t}_i = \mathbf{Y}_i^T\mathbf{w}_i^*$, the covariance (not correlation) between each latent variable and the transformed KPIs (along $\mathbf{v}_i^*, i = 1, \cdots, \gamma$) is maximized. In the next step, PLS finds the least squares (LS) solution from latent variables to KPIs. It is remarkable that $\mathbf{R}^T\mathbf{YY}^T\mathbf{R}$ is a diagonal matrix, which means that the inversion is computed quite efficiently.

Alternatively, in order to eliminate the problem caused by collinearity, the following

SVD is carried out[1]

$$\frac{\mathbf{Y}\mathbf{Y}^T}{N-1} = \begin{bmatrix}\mathbf{P}_1 & \mathbf{P}_2\end{bmatrix}\begin{bmatrix}\mathbf{\Lambda}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{\Lambda}_2\end{bmatrix}\begin{bmatrix}\mathbf{P}_1^T\\ \mathbf{P}_2^T\end{bmatrix}, \tag{3.7}$$
$$\mathbf{\Lambda}_1 = \text{diag}(\lambda_1^2, \cdots, \lambda_{\bar{m}}^2),\ \mathbf{\Lambda}_2 = \text{diag}(\lambda_{\bar{m}+1}^2, \cdots, \lambda_m^2),$$
$$\lambda_1^2 \geq \cdots \geq \lambda_{\bar{m}}^2 >> \lambda_{\bar{m}+1}^2 \geq \cdots \geq \lambda_m^2,$$

which is equivalent to PCA for dimension reduction. To better explain the basic idea, a 3-dimensional numerical example is given. The normalized data are plotted in Figure 3.1-a. There are three process variables, y_1 and y_2 are multivariate normal distributed, y_3 is a linear combination of them and is contaminated by noise. $\boldsymbol{\theta}$ is a linear combination of these three variables. By performing the SVD (3.7), the directions of data variability, $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ (define $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$), can be extracted in the descending order. Their bilateral relationships are shown in Figure 3.1-b ('o'). We can see that the most significant variability occurs in the '$\mathbf{p}_1 - \mathbf{p}_2$' plane, the variability along $\mathbf{p}_3$ is relatively much smaller and is caused by the process and measurement noise. Thus it is reasonable to represent the original data with a lower dimension. The red 'x's in Figure 3.1-b denote the denoised data (Note that different from this illustrative example, the variability along $\mathbf{p}_3$, which causes rank deficiency of $\mathbf{Y}\mathbf{Y}^T$, is quite close to zero). It can be seen that the two data sets (original and projected) contain quite similar information.

Similar to the PLS solution (3.6), the coefficient matrix $\mathbf{\Psi}$ can be alternatively estimated as

$$\hat{\mathbf{\Psi}} = \bar{\mathbf{\Psi}}\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T,\ \bar{\mathbf{\Psi}} = \frac{\mathbf{\Theta}\bar{\mathbf{Y}}^T}{N-1},\ \bar{\mathbf{Y}} = \mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T\mathbf{Y}. \tag{3.8}$$

The estimation performance is equivalent to the principal component regression, nevertheless, it provides superior monitoring performance to either PCA- or PLS-based approaches. The process variable space has been divided into two orthogonal subspaces $\mathcal{S}_{\hat{\mathbf{y}}} = span\{\mathbf{P}_1\} \subset \mathcal{R}^{\bar{m}}$ and $\mathcal{S}_{\tilde{\mathbf{y}}} = span\{\mathbf{P}_2\} \subset \mathcal{R}^{m-\bar{m}}$ where the former contains informative part while the latter is the residual part which contains no useful information for the process variables themselves, thus not useful for the KPIs either. From the monitoring point of view, the denoised process variable data $\bar{\mathbf{Y}}$ (represented in a reduced order coordinate system) contain both KPI-related and -unrelated information. In order to identify the KPI-related and -unrelated subspaces in the new coordinate system, the following

[1]The determination of $\bar{m}$ plays an important role for an optimal fault detection system design. It should be selected in such a way that $\mathbf{\Lambda}_2$ contains those "zero" components that are numerically quite sensitive to the inversion computation.

SVD is done on $\frac{\mathbf{\Theta}\bar{\mathbf{Y}}^T}{N-1}$

$$\frac{\mathbf{\Theta}\bar{\mathbf{Y}}^T}{N-1} = \mathbf{Q}_{\boldsymbol{\theta}} \begin{bmatrix} \mathbf{R} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{S}_1^T \\ \mathbf{S}_2^T \end{bmatrix}.$$

As a result, $\mathcal{S}_{\boldsymbol{\theta}} = span\{\mathbf{S}_1\}$ represents the KPI-related subspace while $\mathcal{S}_{\boldsymbol{\theta}^\perp} = span\{\mathbf{S}_2\}$ represents the KPI-unrelated subspace in the new coordinate system. As illustrated in Figure 3.1-c and Figure 3.1-d, $\mathbf{S}_1$ denotes the direction of $\bar{\mathbf{Y}}$ that is mostly correlated with the KPIs ($\text{corr}(\mathbf{S}_1^T\bar{\mathbf{y}}, \boldsymbol{\theta}) = 0.9997$) while $\mathbf{S}_2$ denotes the orthogonal direction that is not useful for the KPIs ($\text{corr}(\mathbf{S}_2^T\bar{\mathbf{y}}, \boldsymbol{\theta}) = -0.0247$). Based on these decompositions, the complete process variable space can be monitored.

As in the revised PLS-based monitoring scheme, to monitor the KPI-related process variable subspace, it is reasonable to establish the following test statistic

$$T^2_{\hat{\boldsymbol{\theta}}} = \mathbf{y}^T\mathbf{P}_1\mathbf{\Lambda}_1^{-1/2}\mathbf{S}_1\mathbf{S}_1^T\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T\mathbf{y}. \tag{3.9}$$

Since $\mathbf{S}_1^T\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, thus with sufficient training data, the threshold can be set as

$$J_{th,T^2_{\hat{\boldsymbol{\theta}}}} = \chi^2_{1-\alpha}(l)$$

For monitoring the KPI-unrelated part, we do the following transformation

$$\boldsymbol{\theta}^\perp = \begin{bmatrix} \lambda_m\mathbf{S}_2^T\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T \\ \mathbf{\Xi}^{1/2}\mathbf{P}_2^T \end{bmatrix}\mathbf{y}$$

where $\mathbf{\Xi} = diag(\frac{\lambda_m^2}{\lambda_{m+1}^2}, \cdots, \frac{\lambda_m^2}{\lambda_{m-1}^2}, 1)$ is utilized to increase the numerical robustness regarding to the very small λ_m.

Since $E(\boldsymbol{\theta}^\perp(\boldsymbol{\theta}^\perp)^T) = \lambda_m^2\mathbf{I}_{(m-l)\times(m-l)}$, the following test statistic can be established to monitor the part of process variables that is useless for the KPIs

$$T^2_{\boldsymbol{\theta}^\perp} = \mathbf{y}^T\left(\lambda_m^2\mathbf{P}_1\mathbf{\Lambda}_1^{-1/2}\mathbf{S}_2\mathbf{S}_2^T\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T + \mathbf{P}_2\mathbf{\Xi}\mathbf{P}_2^T\right)\mathbf{y} \sim \lambda_m^2\chi^2(m-l). \tag{3.10}$$

The corresponding threshold $J_{th,T^2_{\boldsymbol{\theta}^\perp}}$ can be determined using the χ^2-distribution for a given significance level α.

Design procedures of the alternative KPI monitoring scheme is given in Algorithm 3.2.

In addition to the process variables, any change in the matrix $\bar{\mathbf{\Psi}}$ can influence the KPIs as well. However, this kind of malfunctions cannot be detected either by (3.9) or (3.10). In order to detect this kind of fault, the online KPI measurement is required. Since $E(\boldsymbol{\theta}\boldsymbol{\theta}^T) = E((\hat{\boldsymbol{\theta}} + \mathbf{e}_{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}} + \mathbf{e}_{\boldsymbol{\theta}})^T) = E(\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}^T) + E(\mathbf{e}_{\boldsymbol{\theta}}\mathbf{e}_{\boldsymbol{\theta}}^T)$, we have $\mathbf{e}_{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\theta}} - \mathbf{Q}_{\boldsymbol{\theta}}\mathbf{R}^2\mathbf{Q}_{\boldsymbol{\theta}}^T)$ where $\Sigma_{\boldsymbol{\theta}} \approx \frac{\mathbf{\Theta}\mathbf{\Theta}^T}{N-1}$. Thus the following test statistic is established

$$T^2_{\boldsymbol{\theta}_\mathbf{e}} = (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\mathbf{U}\mathbf{\Omega}\mathbf{U}^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \sim d_l^2\chi^2(l) \tag{3.11}$$

Algorithm 3.2. *An alternative static KPI monitoring scheme*

Based on the normalized data $\mathbf{Y} \in \mathcal{R}^{m\times N}$ *and* $\mathbf{\Theta} \in \mathcal{R}^{l\times N}$,

S1: Do an SVD on $\frac{\mathbf{Y}\mathbf{Y}^T}{N-1} \Longrightarrow \mathbf{P}_1, \mathbf{\Lambda}_1; \mathbf{P}_2, \mathbf{\Xi}, \lambda_m^2$.

S2: Do another SVD on $\bar{\mathbf{\Psi}} = \frac{\mathbf{\Theta}(\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T\mathbf{Y})^T}{N-1} \Longrightarrow \mathbf{Q}_\theta, \mathbf{R}, \mathbf{S}_1^T; \mathbf{S}_2^T$.

S3: Compute the thresholds $J_{th,T^2_{\hat{\theta}}}$ *and* $J_{th,T^2_{\theta^\perp}}$.

Using the normalized online data $\mathbf{y}(k)$,

S4: Build the statistics $T^2_{\hat{\theta}}$ *and* $T^2_{\theta^\perp}$ *according to (3.9) and (3.10).*

S5: Check the decision logic:

$$\begin{cases} T^2_{\hat{\theta}} > J_{th,T^2_{\hat{\theta}}} \text{ and } T^2_{\theta^\perp} \le J_{th,T^2_{\theta^\perp}} \Rightarrow & \textit{fault influences KPI} \\ T^2_{\hat{\theta}} \le J_{th,T^2_{\hat{\theta}}} \text{ and } T^2_{\theta^\perp} > J_{th,T^2_{\theta^\perp}} \Rightarrow & \textit{fault does not influence KPI} \\ T^2_{\hat{\theta}} > J_{th,T^2_{\hat{\theta}}} \text{ and } T^2_{\theta^\perp} > J_{th,T^2_{\theta^\perp}} \Rightarrow & \textit{both kinds of faults happen} \\ T^2_{\hat{\theta}} \le J_{th,T^2_{\hat{\theta}}} \text{ and } T^2_{\theta^\perp} \le J_{th,T^2_{\theta^\perp}} \Rightarrow & \textit{fault-free.} \end{cases}$$

where

$$\mathbf{\Omega} = diag(\frac{d_l^2}{d_1^2}, \cdots, \frac{d_l^2}{d_{l-1}^2}, 1)$$

with

$$\mathbf{\Sigma}_\theta - \mathbf{Q}_\theta \mathbf{R}^2 \mathbf{Q}_\theta^T = \mathbf{U}\mathbf{D}\mathbf{U}^T,\ \mathbf{D} = diag(d_1^2, \cdots, \lambda_{d_l}^2),\ d_1^2 \ge \cdots \ge d_l^2.$$

Remark: Among these test statistics, in the case that the KPIs are not online measurable,

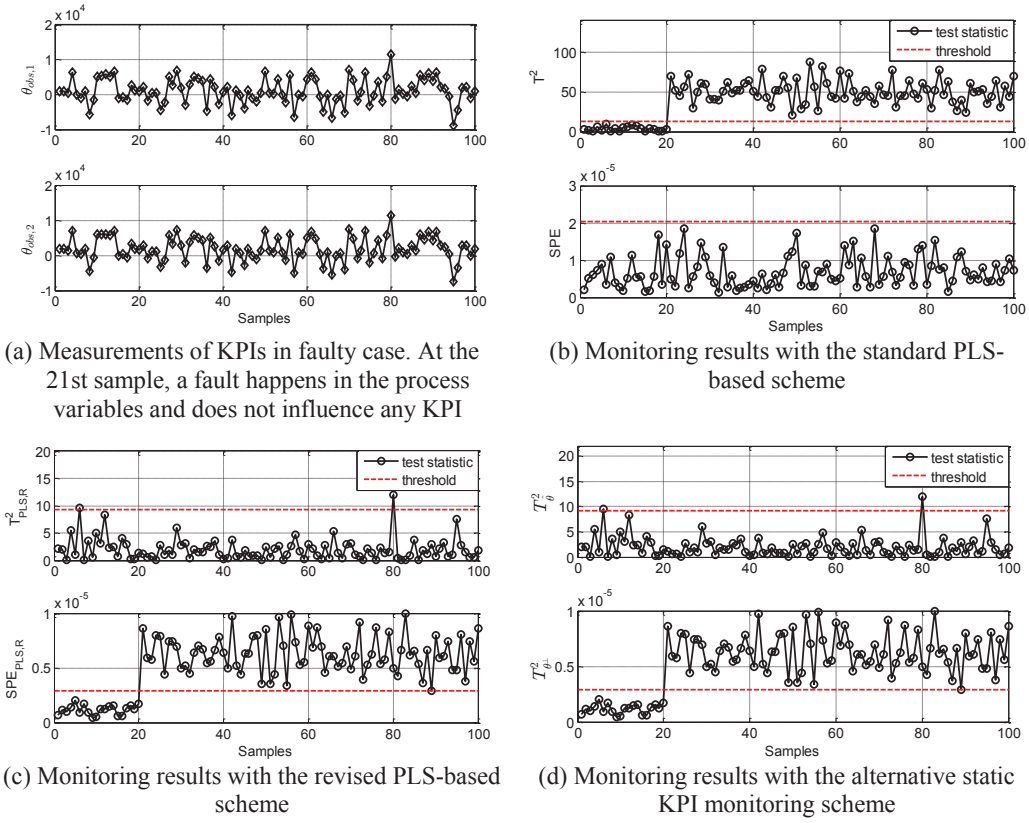
- $T^2_{\hat{\theta}}$ can detect the KPI-related faults in the process variables and contains no useless information for the KPIs.
- $T^2_{\theta^\perp}$ is able to detect the KPI-unrelated faults in the process variables.

Compared to the PLS-based schemes, the design effort of this alternative method is quite low. No complicated computations like the cross validation is required. Moreover, two test statistics are involved and their monitoring and diagnosis performance is improved over the standard algorithm. In addition, in the case that KPIs are online measurable, the faults happened in $\bar{\mathbf{\Psi}}$, which cannot be detected by (3.9) or (3.10), can be detected by the $T^2_{\theta_e}$ test statistic. A simulation example is given in the next section to demonstrate this scheme.

3.4 Numerical examples

In this section, an open-loop synthetic numerical example is considered:

$$\mathbf{y}_{obs}(k) = \mathbf{W}\mathbf{z}(k) + \mathbf{\eta}(k)$$
$$\mathbf{\theta}_{obs}(k) = \mathbf{\Psi}_{obs}\mathbf{y}_{obs}(k) + \mathbf{b}_{obs} + \mathbf{\xi}(k)$$



(a) Measurements of KPIs in faulty case. At the 21st sample, a fault happens in the process variables and does not influence any KPI

(b) Monitoring results with the standard PLS-based scheme

(c) Monitoring results with the revised PLS-based scheme

(d) Monitoring results with the alternative static KPI monitoring scheme

Figure 3.2: Comparison of monitoring results from static schemes

where

$$\mathbf{W} = \text{rand}(m, n), \mathbf{\Psi}_{obs} = 1 + \text{rand}(l, m), \mathbf{b}_{obs} = [1055, 1825]^T$$

are model parameters with $n = 3, m = 15$ and $l = 2$ denoting the dimensions of $\mathbf{z}, \mathbf{y}_{obs}$ and $\boldsymbol{\theta}_{obs}$, respectively. To model the steady state of a process, $\mathbf{z}$ is simulated by random number sequences, *i.e.* $\mathbf{z}_1 \sim \mathcal{N}(0, 4), \mathbf{z}_2 \sim \mathcal{N}(0, 1.96)$ and $\mathbf{z}_3 \sim \mathcal{N}(0, 1.44)$. In addition, model uncertainties and measurement noises are simulated by $\eta_i \sim \mathcal{N}(0, 1e-6), i = 1, \cdots, m$, and $\xi_i \sim \mathcal{N}(0, 1e-8), i = 1, \cdots, l$, respectively. One hundred samples are generated to train the model. In addition, a deterministic fault is injected into the process variable space at the 21st sample.

As depicted in Figure 3.2-a, the fault is selected in such a way that it changes the latent variables but does not influence any KPI. Monitoring results of the standard PLS-based scheme is given in Figure 3.2-b. We can see that the T^2 statistic detects the fault while SPE cannot, which is not as expected (with significance level $\alpha = 1\%$). In comparison, the monitoring results of the revised PLS-based scheme and the alternative

Table 3.1: Comparison of offline computation time ('seconds')

Standard PLS-based scheme	Revised PLS-based scheme	The alternative scheme
0.8818	0.8665	0.0056

static scheme are given in Figure 3.2-c and Figure 3.2-d, respectively. As expected, the $T^2_{PLS,R}$ and $T^2_{\hat{\theta}}$ statistics, which are responsible for KPI-related part, have not detected the fault. The $SPE_{PLS,R}$ and $T^2_{\theta^\perp}$ statistics, which are responsible for KPI-unrelated part, have successfully detected the fault. These examples demonstrate the performance improvements of proposed schemes over the standard one. Regarding the engineering effort, the offline computation time is measured in MATLAB. Due to various hardware and software configurations, the absolute time makes no sense. But the time consumed under the same condition is reasonable for comparing different monitoring schemes. Table 3.4 lists the values, from which we can see that the standard and revised PLS-based schemes (*leave-one-out* for cross validation) consume comparable computation resources while the alternative scheme is significantly faster. Moreover, the standard PLS-based scheme uses a bit more time than the revised PLS-based scheme. The reason it the calculation of the threshold for the SPE statistic.

3.5 Concluding remarks

This chapter focuses on the data-driven monitoring schemes for static processes. Motivated by the drawbacks of the standard PLS-based process monitoring scheme, a revised approach based on the PLS algorithm is firstly proposed. It can correctly detect the KPI-related and -unrelated faults. Although similar modified approaches have been recently reported by other researchers [93, 136], this approach uses much less test statistics and thus is more appealing for practical applications. In order to reduce the engineering effort, an alternative static KPI monitoring scheme is proposed subsequently. This scheme provides improved monitoring performance (over the standard PLS-based scheme) and fulfils all the requirements listed in the *problem formulation*. Nevertheless, the application scope of these static schemes is limited to static processes. Dynamic monitoring approaches will be addressed in the next chapter.

4 Data-driven KPI monitoring techniques for lumped-parameter processes

Although KPI monitoring schemes designed for static processes have received great success in industry, their application scope is restricted. Those methods are simple and suitable for the periods when the processes are steady, *i.e.* the mean vectors of the process variables and the KPIs are quasi-constant. In practice, however, the mean vector of the process variables could be time-varying *e.g.* due to continuous supervisory control actions, especially in large-scale systems. In order to monitor such processes, the inherent dynamics must be effectively taken into account. As discussed in Chapters 1 and 2, the PS-based techniques, the FDF, and the DO are powerful dynamic monitoring tools. A requirement of all these approaches is that a reliable mathematical description of the process dynamics be available. Derivation of mathematical models based on first principles can be costly or time-consuming, particularly in the process industry.

Motivated by these observations, data-driven design of model-based process monitoring systems has attracted much research interest. Strongly stimulated by the development of system identification techniques, the two-step scheme "system identification + model-based monitoring system design" has become a standard one and been applied on various benchmark processes. Nevertheless, according to our recent research activities [28], the design effort can be largely reduced by the direct identification of the process monitoring system. In this chapter, we will present some new results. The starting point is a residual generator that we wish to identify. Based on the assumption that the residual, which is a combination of the process and measurement noise, is uncorrelated with the process variables and the past KPIs, its covariance matrix, which is essential for residual evaluation, can be readily extracted from the process I/O data. Then by eliminating the influences of noise from the process I/O data, residual generators for measurable KPIs and unmeasurable KPIS can be directly identified.

4.1 Preliminaries and objective

Assuming that the dynamics from the low-level process variables to the high-level KPIs can be modelled by (2.2-2.3), we want to design the following residual generator

$$\hat{\mathbf{z}}(k+1) = \mathbf{A}\hat{\mathbf{z}}(k) + \mathbf{B}\mathbf{y}(k) + \mathbf{L}\mathbf{r}(k), \hat{\mathbf{z}}(0) = \mathbf{0}, \tag{4.1}$$

$$\mathbf{r}(k) = \boldsymbol{\theta}(k) - \mathbf{C}\hat{\mathbf{z}}(k) - \mathbf{D}\mathbf{y}(k). \tag{4.2}$$

Here $\mathbf{r}(k)$ denotes a primary residual signal and can be alternatively written as

$$\begin{aligned}\mathbf{r} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} &= \boldsymbol{\theta} - \mathbf{C}(p\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1}\left(\mathbf{L}\boldsymbol{\theta} + (\mathbf{B} - \mathbf{L}\mathbf{D})\mathbf{y}\right) - \mathbf{D}\mathbf{y} \\ &= \left(-\mathbf{C}(p\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1}\mathbf{L} + \mathbf{I}\right)\boldsymbol{\theta} \\ &\quad - \left(\mathbf{C}(p\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1}(\mathbf{B} - \mathbf{L}\mathbf{D}) + \mathbf{D}\right)\mathbf{y} \\ &:= \begin{bmatrix} -\hat{\mathbf{N}}(p) & \hat{\mathbf{M}}(p) \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\theta} \end{bmatrix}\end{aligned} \tag{4.3}$$

where p denotes the *z-transformation* operator, $\hat{\mathbf{M}}(p) = -\mathbf{C}(p\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1}\mathbf{L} + \mathbf{I}$, and $\hat{\mathbf{N}}(p) = \mathbf{C}(p\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1}(\mathbf{B} - \mathbf{L}\mathbf{D}) + \mathbf{D}$. The above residual generator is defined as the *kernel representation* [23] and builds the basis for KPI monitoring. In the model-based framework, the core of designing the kernel representation is to select an appropriate observer gain matrix $\mathbf{L}$ because the other parameters are given by first principle models. Differently, in the data-driven framework, the major objective is to design the whole dynamic residual generator based on the normal process I/O data. In addition, data-driven design of a residual evaluation system is another objective.

4.2 Construction of the I/O data model

Before constructing the data-based process model, assume

$$E(\mathbf{r}(k)\mathbf{y}^T(k-i)) = 0, \ \forall\ i = -s, \cdots, -1, 0, 1, \cdots, s_p \tag{4.4}$$

$$E(\mathbf{r}(k)\boldsymbol{\theta}^T(k-j)) = 0, \ \forall\ j = 1, 2, \cdots, s_p. \tag{4.5}$$

The first assumption means that the residual vector is uncorrelated with the low-level process variables, which is quite reasonable in the *open-loop configuration*. The second assumption indicates that the residual signals are uncorrelated with the past KPI measurements, which can be generally guaranteed by the feature of residual generators (*e.g.* Kalman filter generates white residuals that are uncorrelated with past data).

Based on the residual generator (4.1-4.2), an I/O data model can be constructed as

$$\boldsymbol{\Theta}_{k,k+s} = \boldsymbol{\Gamma}_s\hat{\mathbf{Z}}_k + \mathbf{H}_{y,s}\mathbf{Y}_{k,k+s} + \mathbf{H}_{r,s}\mathbf{R}_{k,k+s} \tag{4.6}$$

where $\mathbf{\Theta}_{k,k+s}$ is built from $\boldsymbol{\theta}(k), k = 1, \cdots, k+s+N_c-1$ as

$$\mathbf{\Theta}_{k,k+s} = \begin{bmatrix} \boldsymbol{\theta}(k) & \cdots & \boldsymbol{\theta}(k+N_c-1) \\ \vdots & \vdots & \vdots \\ \boldsymbol{\theta}(k+s) & \cdots & \boldsymbol{\theta}(k+s+N_c-1) \end{bmatrix} \in \mathcal{R}^{(s+1)l \times N_c},$$

$\mathbf{Y}_{k,k+s} \in \mathcal{R}^{(s+1)m \times N_c}$ and $\mathbf{R}_{k,k+s} \in \mathcal{R}^{(s+1)l \times N_c}$ are built in the same way as $\mathbf{\Theta}_{k,k+s}$, $\hat{\mathbf{Z}}_k$ is built as

$$\hat{\mathbf{Z}}_k = \begin{bmatrix} \hat{\mathbf{z}}(k) & \cdots & \hat{\mathbf{z}}(k+N_c-1) \end{bmatrix} \in \mathcal{R}^{n \times N_c}, \tag{4.7}$$

$\mathbf{\Gamma}_s$ is represented by $\mathbf{C}$ and $\mathbf{A}$ as

$$\mathbf{\Gamma}_s = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^s \end{bmatrix} \in \mathcal{R}^{(s+1)l \times n}, \tag{4.8}$$

$\mathbf{H}_{y,s}$ is represented by $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ as

$$\mathbf{H}_{y,s} = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{CA}^{s-1}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & \mathbf{D} \end{bmatrix} \in \mathcal{R}^{(s+1)l \times (s+1)m}, \tag{4.9}$$

$\mathbf{H}_{r,s} \in \mathcal{R}^{(s+1)l \times (s+1)l}$ is represented by $(\mathbf{A}, \mathbf{L}, \mathbf{C}, \mathbf{I})$ in the same style as $\mathbf{H}_{y,s}$.

Similarly to (4.6), the I/O data model can be written as

$$\begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \mathbf{\Theta}_{k,k+s} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{H}_{y,s} & \mathbf{\Gamma}_s \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \hat{\mathbf{Z}}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{H}_{r,s}\mathbf{R}_{k,k+s} \end{bmatrix}. \tag{4.10}$$

Let $\mathbf{\Psi}_s := \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{H}_{y,s} & \mathbf{\Gamma}_s \end{bmatrix} \in \mathcal{R}^{(s+1)(m+l) \times ((s+1)m+n)}$, if there exists a matrix $\mathbf{\Psi}_s^{\perp} \in \mathcal{R}^{((s+1)l-n) \times (s+1)(m+l)}$ such that $\mathbf{\Psi}_s^{\perp}\mathbf{\Psi}_s = \mathbf{0}$, then similar to the kernel representation (4.3),

$$\mathbf{R}_s = \mathbf{\Psi}_s^{\perp} \begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \mathbf{\Theta}_{k,k+s} \end{bmatrix} = \mathbf{\Psi}_s^{\perp} \begin{bmatrix} \mathbf{0} \\ \mathbf{H}_{r,s}\mathbf{R}_{k,k+s} \end{bmatrix} \tag{4.11}$$

is a kernel representation of the I/O model (4.10) as well, which maps the process and KPI data onto multiple residual signals.

4.3 Identification of the kernel representation

Compared to (4.3) where the dynamic information is embedded in the kernel parameter functions $\hat{\mathbf{N}}(p)$ and $\hat{\mathbf{M}}(p)$, in the data-based kernel representation (4.11), the I/O data are stacked columnwise such that the dynamic information can be numerically extracted. The data-based kernel $\mathbf{\Psi}_s^{\perp}$ is a constant matrix to be identified.

Note that in the fault-free case, the primary residual $\mathbf{R}_{k,k+s}$ consists of the process and measurement noise. To reveal the basic idea behind identification, let us first consider the noise-free case, *i.e.*

$$\begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \mathbf{\Theta}_{k,k+s} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{H}_{y,s} & \mathbf{\Gamma}_s \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \hat{\mathbf{Z}}_k \end{bmatrix}. \tag{4.12}$$

Assume that $\begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \hat{\mathbf{Z}}_k \end{bmatrix}$ has full row rank, then we know that the left null space of $\begin{bmatrix} \mathbf{Y}_{k,k+s} \\ \mathbf{\Theta}_{k,k+s} \end{bmatrix}$ is identical to the left null space of $\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{H}_{y,s} & \mathbf{\Gamma}_s \end{bmatrix}$. Thus the kernel representation of the data-based form can be identified from the process and KPI data using *e.g.* an SVD. However, in practice noise is inevitable since the process and KPI data sets are contaminated by noise. Neglecting the noise component will decrease the identification quality and further the performance of the monitoring system. To lower the influence of noise on identification, let us build another extended state equation as

$$\hat{\mathbf{Z}}_k = (\mathbf{A} - \mathbf{LC})^{s_p} \hat{\mathbf{Z}}_{k-s_p} + \mathbf{\Phi}_{y,s_p} \mathbf{Y}_{k-s_p,k-1} + \mathbf{\Phi}_{\theta,s_p} \mathbf{\Theta}_{k-s_p,k-1} \tag{4.13}$$

where

$$\mathbf{\Phi}_{y,s_p} = \begin{bmatrix} (\mathbf{A} - \mathbf{LC})^{s_p-1}(\mathbf{B} - \mathbf{LD}) & \cdots & (\mathbf{A} - \mathbf{LC})(\mathbf{B} - \mathbf{LD}) & (\mathbf{B} - \mathbf{LD}) \end{bmatrix} \in \mathcal{R}^{n \times s_p m}, \tag{4.14}$$

$$\mathbf{\Phi}_{\theta,s_p} = \begin{bmatrix} (\mathbf{A} - \mathbf{LC})^{s_p-1}\mathbf{L} & \cdots & (\mathbf{A} - \mathbf{LC})\mathbf{L} & \mathbf{L} \end{bmatrix} \in \mathcal{R}^{n \times s_p l}, \tag{4.15}$$

$\mathbf{Y}_{k-s_p,k-1}$ and $\mathbf{\Theta}_{k-s_p,k-1}$ are built from $\mathbf{y}(i), i = k - s_p, \cdots, k + N_c - 2$, and $\mathbf{\theta}(i), i = k - s_p, \cdots, k + N_c - 2$, in the same way as $\mathbf{\Theta}_{k,k+s}$.

Since the basic requirement for observer design is *stability*, *i.e.* all eigenvalues of $(\mathbf{A} - \mathbf{LC})$ are inside the unit circle. As a result for a sufficiently large integer s_p, we have $(\mathbf{A} - \mathbf{LC})^{s_p} \approx \mathbf{0}$, which makes

$$\hat{\mathbf{Z}}_k \approx \begin{bmatrix} \mathbf{\Phi}_{y,s_p} & \mathbf{\Phi}_{\theta,s_p} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{k-s_p,k-1} \\ \mathbf{\Theta}_{k-s_p,k-1} \end{bmatrix} := \mathbf{\Phi}_{s_p} \mathbf{\Omega}_{k-s_p,k-1} \tag{4.16}$$

Substituting (4.16) into (4.6) gives

$$\boldsymbol{\Theta}_{k,k+s} \approx \boldsymbol{\Gamma}_s \boldsymbol{\Phi}_{s_p} \boldsymbol{\Omega}_{k-s_p,k-1} + \mathbf{H}_{y,s} \mathbf{Y}_{k,k+s} + \mathbf{H}_{r,s} \mathbf{R}_{k,k+s} \tag{4.17}$$

$$:= \begin{bmatrix} \boldsymbol{\Gamma}_s \boldsymbol{\Phi}_{s_p} & \mathbf{H}_{y,s} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix} + \mathbf{H}_{r,s} \mathbf{R}_{k,k+s}. \tag{4.18}$$

In the above representation, $\boldsymbol{\Theta}_{k,k+s}$ and $\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}$ are known from the process I/O data. $\mathbf{H}_{r,s}\mathbf{R}_{k,k+s}$ represents the noise component and is unknown. We expect to identify $\begin{bmatrix} \boldsymbol{\Gamma}_s \boldsymbol{\Phi}_{s_p} & \mathbf{H}_{y,s} \end{bmatrix}$ using a regular left transformation. By projecting $\boldsymbol{\Theta}_{k,k+s}$ onto the orthogonal subspace of the row subspace of $\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}$, we have

$$\begin{aligned}
&\mathrm{proj}_{\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}}{}^{\perp} \boldsymbol{\Theta}_{k,k+s} \\
&:= \boldsymbol{\Theta}_{k,k+s} \left(\mathbf{I} - \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}^T \left(\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}^T \right)^{-1} \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix} \right) \\
&= \mathbf{H}_{r,s}\mathbf{R}_{k,k+s} \left(\mathbf{I} - \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}^T \left(\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}^T \right)^{-1} \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix} \right) \\
&= \mathbf{H}_{r,s}\mathbf{R}_{k,k+s}
\end{aligned} \tag{4.19}$$

where the last step is due to the assumptions made in (4.4-4.5).

Now do the following LQ-decomposition

$$\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \\ \boldsymbol{\Theta}_{k,k+s} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{L}_{23} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \\ \mathbf{Q}_3^T \end{bmatrix}. \tag{4.20}$$

Since $\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}$ is full rank, the null subspace of $\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}$ is identical to the null subspace of $\begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix}$ $\left(\text{note } \begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \right)$, which is spanned by the rows of $\mathbf{Q}_3^T$. As a result, the projection of $\boldsymbol{\Theta}_{k,k+s}$ onto the orthogonal subspace of the row subspace of $\begin{bmatrix} \boldsymbol{\Omega}_{k-s_p,k-1} \\ \mathbf{Y}_{k,k+s} \end{bmatrix}$ is identical to the projection of $\boldsymbol{\Theta}_{k,k+s}$ onto the row subspace of

$\mathbf{Q}_3^T$, *i.e.*

$$\begin{aligned}&\text{proj}_{\begin{bmatrix}\boldsymbol{\Omega}_{k-s_p,k-1}\\ \mathbf{Y}_{k,k+s}\end{bmatrix}}{}^{\perp}\boldsymbol{\Theta}_{k,k+s} := \boldsymbol{\Theta}_{k,k+s}\left(\mathbf{Q}_3(\mathbf{Q}_3^T\mathbf{Q}_3)^{-1}\mathbf{Q}_3^T\right) = \mathbf{L}_{33}\mathbf{Q}_3^T\\ \Longrightarrow\quad &\mathbf{H}_{r,s}\mathbf{R}_{k,k+s} = \mathbf{L}_{33}\mathbf{Q}_3^T.\end{aligned} \tag{4.21}$$

Thus based on the LQ-decomposition (4.20), we have

$$\begin{bmatrix}\mathbf{Y}_{k,k+s}\\ \boldsymbol{\Theta}_{k,k+s}\end{bmatrix} = \begin{bmatrix}\mathbf{L}_{21} & \mathbf{L}_{22}\\ \mathbf{L}_{31} & \mathbf{L}_{23}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_2^T\\ \mathbf{Q}_3^T\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{H}_{r,s}\mathbf{R}_{k,k+s}\end{bmatrix} \tag{4.22}$$

where $\begin{bmatrix}\mathbf{L}_{21} & \mathbf{L}_{22}\\ \mathbf{L}_{31} & \mathbf{L}_{23}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_2^T\\ \mathbf{Q}_3^T\end{bmatrix}$ is the denoised process and KPI data. Further since $\begin{bmatrix}\mathbf{Q}_2^T\\ \mathbf{Q}_3^T\end{bmatrix}$ is of full row rank, the left null subspace of $\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{H}_{y,s} & \boldsymbol{\Gamma}_s\end{bmatrix}$ is identical to the left null subspace of $\begin{bmatrix}\mathbf{L}_{21} & \mathbf{L}_{22}\\ \mathbf{L}_{31} & \mathbf{L}_{23}\end{bmatrix}$. By an SVD

$$\begin{bmatrix}\mathbf{L}_{21} & \mathbf{L}_{22}\\ \mathbf{L}_{31} & \mathbf{L}_{23}\end{bmatrix} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_1 & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Lambda}_2\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^T\\ \mathbf{V}_2^T\end{bmatrix}, \tag{4.23}$$

we have

$$\boldsymbol{\Lambda}_2 \approx \mathbf{0} \implies \boldsymbol{\Psi}_s^{\perp} = \mathbf{P}\mathbf{U}_2^T \in \mathcal{R}^{((s+1)l-n)\times(s+1)(m+l)} \tag{4.24}$$

where $\mathbf{P} \in \mathcal{R}^{((s+1)l-n)\times((s+1)l-n)}$ denotes the regular transformation matrix. Subsequently, for simplicity of notation, we assume that $\mathbf{P} = \mathbf{I}$. At this stage, the data-based kernel representation (4.11) has been realized based only on the normal process and KPI data, *i.e.*

$$\mathbf{Res} = \boldsymbol{\Psi}_s^{\perp}\begin{bmatrix}\mathbf{Y}_{k,k+s}\\ \boldsymbol{\Theta}_{k,k+s}\end{bmatrix} := \begin{bmatrix}-\boldsymbol{\Psi}_{s,y}^{\perp} & \boldsymbol{\Psi}_{s,\boldsymbol{\theta}}^{\perp}\end{bmatrix}\begin{bmatrix}\mathbf{Y}_{k,k+s}\\ \boldsymbol{\Theta}_{k,k+s}\end{bmatrix} = \boldsymbol{\Psi}_{s,\boldsymbol{\theta}}^{\perp}\mathbf{L}_{33}\mathbf{Q}_3^T \tag{4.25}$$

where $\boldsymbol{\Psi}_{s,y}^{\perp} = -\boldsymbol{\Psi}_s^{\perp}(:, 1:(s+1)m)$ and $\boldsymbol{\Psi}_{s,\boldsymbol{\theta}}^{\perp} = \boldsymbol{\Psi}_s^{\perp}(:, (s+1)m+1:(s+1)(m+l))$. It is important to mention that except for the parameters of the kernel representation (4.11), the covariance matrix of the extended form of the primary residual is simultaneously identified, which is $\mathbf{L}_{33}\mathbf{L}_{33}^T/(N_c-1)$.

Algorithm 4.1. *Data-driven design of kernel representation based residual generator*

S1: Select design parameters s, s_p and construct process I/O data matrices $\mathbf{\Omega}_{k-s_p,k-1}$, $\mathbf{Y}_{k,k+s}$ and $\mathbf{\Theta}_{k,k+s}$.

S2: Perform an LQ-decomposition (4.20) and then an SVD (4.23).

S3: Extract the kernel representation matrices $\mathbf{\Psi}_{s,y}^{\perp}$ and $\mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp}$.

S4: Determine the thresholds from (4.28), (4.30), (4.31).

Based on the stacked online data $\mathbf{y}_s(k)$ and $\boldsymbol{\theta}_s(k)$,

S5: Build the test statistic T^2 or SPE or T_r^2.

S6: Check the decision logic:

$$\begin{cases} \textit{test statistic} > \textit{threshold} & \Rightarrow \textit{faulty} \\ \textit{otherwise} & \Rightarrow \textit{fault-free.} \end{cases}$$

4.4 Kernel representation based KPI monitoring

4.4.1 Parity space based residual generation

Denote $\boldsymbol{\theta}_s(k) = [\boldsymbol{\theta}^T(k), \cdots, \boldsymbol{\theta}^T(k+s)]^T$ and $\mathbf{y}_s(k) = [\mathbf{y}^T(k), \cdots, \mathbf{y}^T(k+s)]^T$. A residual generator can be constructed as follows

$$\mathbf{r}_{(s+1)l-n}(k) = \mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp}\boldsymbol{\theta}_s(k) - \mathbf{\Psi}_{s,y}^{\perp}\mathbf{y}_s(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_r) \tag{4.26}$$

where $\mathbf{\Sigma}_r \approx \frac{\mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp}\mathbf{L}_{33}\mathbf{L}_{33}^T(\mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp})^T}{N_c-1}$.

For residual evaluation, the T^2 test statistic is

$$T^2 = \mathbf{r}_{(s+1)l-n}^T\mathbf{\Sigma}_{\mathbf{r}}^{-1}\mathbf{r}_{(s+1)l-n}, \tag{4.27}$$

and the corresponding threshold is determined by the χ^2-distribution as

$$J_{th,T^2} = \chi_{1-\alpha}^2((s+1)l-n) \tag{4.28}$$

where $(s+1)l-n$ is the degree of freedom and α is the significance level.

However, in practice Σ_r could be rank deficient and cause the standard T^2 to fail. To avoid this problem, we can apply the SPE statistic for residual evaluation:

$$SPE = \mathbf{r}_{(s+1)l-n}^T\mathbf{r}_{(s+1)l-n}. \tag{4.29}$$

Its threshold is determined as

$$J_{th,SPE} = g\chi_{1-\alpha}^2(h) \tag{4.30}$$

where α is the significance level, $g = \frac{\tilde{S}}{2\mu_{SPE}}$ is a scaling factor and $h = \frac{2\mu_{SPE}^2}{\tilde{S}}$ is the degree of freedom of the χ^2-distribution with μ_{SPE} and $\tilde{S}$ being the sample mean and covariance of the SPE statistic.

Alternatively, we can do the following SVD

$$\left(\frac{\mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp}\mathbf{L}_{33}\mathbf{L}_{33}^T(\mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp})^T}{N_c-1}\right) = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T,$$
$$\mathbf{\Lambda} = diag(\lambda_1^2, \cdots, \lambda_{(s+1)l-n}^2),\ \lambda_1^2 \geq \cdots \geq \lambda_{(s+1)l-n}^2.$$

Then with

$$\Omega = \begin{bmatrix} \frac{\lambda_{(s+1)l-n}^2}{\lambda_1^2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \frac{\lambda_{(s+1)l-n}^2}{\lambda_{(s+1)l-n-1}^2} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \frac{\lambda_{(s+1)l-n}^2}{\lambda_{(s+1)l-n}^2}(=1) \end{bmatrix},$$

a revised T^2 statistic can be established as

$$T_r^2 = \mathbf{r}_{(s+1)l-n}^T \mathbf{\Omega} \mathbf{r}_{(s+1)l-n}. \tag{4.31}$$

The threshold can be determined by $J_{th,T_r^2} = \lambda_{(s+1)l-n}^2 \chi_{1-\alpha}^2((s+1)l-n)$.

Algorithm 4.1 summarizes the proposed kernel representation based residual generation scheme.

Remark: From the previous section we know that s is selected much larger than the actual system order n. Based on (4.27) or the other two test statistics, a decision can only be made with $s+1$ available online samples of the process and KPI data. This might result in some obstacles for online monitoring since detection delay is an important criterion for monitoring systems. On the other hand, online computation costs and memory requirements are high. As a result, it is desirable to design alternative monitoring schemes to overcome these difficulties.

4.4.2 Diagnostic observer based residual generation

From Chapter 2, we know that (4.26) is actually a PS-based multiple residual generator. Based on each single residual generator, an equivalent diagnostic observer can be constructed. Let

$$\boldsymbol{\alpha}_s = [\boldsymbol{\alpha}_{s,0}, \boldsymbol{\alpha}_{s,1}, \cdots, \boldsymbol{\alpha}_{s,s}],\ \boldsymbol{\beta}_s = [\boldsymbol{\beta}_{s,0}, \boldsymbol{\beta}_{s,1}, \cdots, \boldsymbol{\beta}_{s,s}] \tag{4.32}$$

where $\boldsymbol{\alpha}_s \in \mathcal{R}^{1\times(s+1)l}$ denotes one parity vector that can for instance be set to any row of $\Psi^{\perp}_{s,\boldsymbol{\theta}}$, $\boldsymbol{\beta}_s \in \mathcal{R}^{1\times(s+1)m}$ or their linear combination. Correspondingly, β_s is obtained from $\Psi^{\perp}_{s,y}$. As described in Chapter 2, we can build an equivalent diagnostic observer from $\boldsymbol{\alpha}_s$ and $\boldsymbol{\beta}_s$ as

$$\mathbf{z}_d(k+1) = \mathbf{G}\mathbf{z}_d(k) + \mathbf{H}y(k) + \mathbf{L}\boldsymbol{\theta}(k) \tag{4.33}$$

$$r(k) = \mathbf{v}\boldsymbol{\theta}(k) - \mathbf{w}\mathbf{z}_d(k) - \mathbf{q}\mathbf{y}(k) \tag{4.34}$$

where $\mathbf{z}_d(k) = \mathbf{T}\mathbf{z}(k) \in \mathcal{R}^s$, and

- $\mathbf{G}$ is stable,
- $\mathbf{TA} - \mathbf{GT} = \mathbf{LC}$, $\mathbf{H} = \mathbf{TB} - \mathbf{LD}$, and
- $\mathbf{vC} - \mathbf{wT} = \mathbf{0}$, $\mathbf{q} = \mathbf{vD}$.

Now define

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \in \mathcal{R}^{s\times s},\ \mathbf{T} = \begin{bmatrix} \boldsymbol{\alpha}_{s,1} & \cdots & \boldsymbol{\alpha}_{s,s-1} & \boldsymbol{\alpha}_{s,s} \\ \boldsymbol{\alpha}_{s,2} & \cdots & \boldsymbol{\alpha}_{s,s} & \mathbf{0} \\ \vdots & \iddots & \iddots & \vdots \\ \boldsymbol{\alpha}_{s,s} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \vdots \\ \mathbf{CA}^{s-2} \\ \mathbf{CA}^{s-1} \end{bmatrix}.$$

It is obvious that the first condition is satisfied since all the eigenvalues of $\mathbf{G}$ are zero. According to $\mathbf{TA} - \mathbf{GT} = \mathbf{LC}$ we can get

$$\mathbf{L} = -\begin{bmatrix} \boldsymbol{\alpha}_0 \\ \vdots \\ \boldsymbol{\alpha}_{s,s-1} \end{bmatrix}. \tag{4.35}$$

Define $\mathbf{w} = [0, \cdots, 0, 1] \in \mathcal{R}^{1\times s}$, from $\mathbf{vC} - \mathbf{wT} = \mathbf{0}$ we know $\mathbf{v} = \boldsymbol{\alpha}_{s,s} \in \mathcal{R}^{1\times l}$. Based on $\mathbf{T}, \mathbf{L}, \mathbf{v}$, $\mathbf{H} = \mathbf{TB} - \mathbf{LD}, \mathbf{q} = \mathbf{vD}$ and the fact that $\boldsymbol{\Psi}^{\perp}_{s,y} = \boldsymbol{\Psi}^{\perp}_{s,\boldsymbol{\theta}}\mathbf{H}_{y,s}$, $\mathbf{H}$ and $\mathbf{q}$ can be constructed from $\boldsymbol{\beta}_s$ as

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{\beta}_{s,0} \\ \vdots \\ \boldsymbol{\beta}_{s,s-1} \end{bmatrix},\ \mathbf{q} = \boldsymbol{\beta}_{s,s}. \tag{4.36}$$

It has been proven in [24] that the parity space based residual generator and the diagnostic observer are equivalent. Based on this result we have $r(k) \sim \mathcal{N}(\mathbf{0}, \frac{\boldsymbol{\alpha}_s \mathbf{L}_{33}\mathbf{L}_{33}^T\boldsymbol{\alpha}_s^T}{N_c-1})$.

Algorithm 4.2. *Kernel representation based diagnostic observer*

S1: *Based on the kernel matrices* $\mathbf{\Psi}_{s,y}^{\perp}$ *and* $\mathbf{\Psi}_{s,\boldsymbol{\theta}}^{\perp}$*, construct the vectors* $\boldsymbol{\alpha}_s$ *and* $\boldsymbol{\beta}_s$*.*

S2: *Select a proper* $\mathbf{g} = [g_1, \cdots, g_s]^T$ *and construct the following matrices:*

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & \cdots & 0 & g_1 \\ 1 & 0 & \dots & 0 & g_2 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & g_{s-1} \\ 0 & \cdots & 0 & 1 & g_s \end{bmatrix}, \begin{cases} \mathbf{q} = \boldsymbol{\beta}_s(sm+1:(s+1)m), \\ \mathbf{H} = \begin{bmatrix} \boldsymbol{\beta}_{s,0} \\ \vdots \\ \boldsymbol{\beta}_{s,s-1} \end{bmatrix} + \mathbf{gq}, \end{cases}$$

$$\begin{cases} \mathbf{v} = \boldsymbol{\alpha}_s(sl+1:(s+1)l), \\ \mathbf{w} = [0, \cdots, 0, 1] \in \mathcal{R}^{1\times s}, \end{cases} \quad \mathbf{L} = -\begin{bmatrix} \boldsymbol{\alpha}_0 \\ \vdots \\ \boldsymbol{\alpha}_{s,s-1} \end{bmatrix} - \mathbf{gv}.$$

S3: *Determine the threshold as* $J_{th,T^2} = \chi^2_{1-\alpha}(1)$*.*

Based on the online data $\mathbf{y}(k)$ *and* $\boldsymbol{\theta}(k)$*,*

S4: *Run the residual generator (4.33-4.34) and evaluation the residual using*

$$T^2 = r(k)^2/\sigma_r^2.$$

S5: *Check the decision logic:*

$$\begin{cases} T^2 > J_{th,T^2} & \Rightarrow \textit{ faulty,} \\ \textit{otherwise} & \Rightarrow \textit{ fault-free.} \end{cases}$$

Similarly to the previous subsection, a T^2-type test statistic can be designed for residual evaluation and the corresponding threshold can be determined using its statistical distribution. In addition, as discussed in Chapter 2, more design freedom can be introduced by adding the last column of $\mathbf{G}$ with the design parameter vector $\mathbf{g} \in \mathcal{R}^s$. Algorithm 4.2 gives the total design procedure for a data-driven DO.

4.4.3 Recursive predictor based residual generation

Both the PS-based residual generator and the DO make use of online KPI measurements. In industrial applications, there are situations where some KPIs are not online/real-time measurable. This section will be devoted to the issue of KPI motoring when they are not measured. We assume the order of the process under consideration is known, and $s = n$. In the single KPI case, we know that $\mathbf{T}$ is invertible ($\mathbf{z}_d(k) = \mathbf{T}\mathbf{z}(k) \in \mathcal{R}^n$) and $v \in \mathcal{R}$. Based on the process description, we can rewrite the DO as

$$\mathbf{z}_d(k+1) = (\mathbf{G} + \mathbf{lcT}^{-1})\mathbf{z}_d(k) + (\mathbf{H} + \mathbf{lD})\mathbf{y}(k) + \mathbf{l}r(k) \tag{4.37}$$

$$\hat{\theta}(k) = \frac{1}{v}\mathbf{w}\mathbf{z}_d(k) + \frac{1}{v}\mathbf{q}\mathbf{y}(k). \tag{4.38}$$

Suppose that the parameters of the kernel representation are given as

$$\boldsymbol{\alpha}_n = [\alpha_{n,0}, \cdots, \alpha_{n,n-1}, 1] \in \mathcal{R}^{1\times(n+1)},\ \boldsymbol{\beta}_n = [\boldsymbol{\beta}_{n,0}, \boldsymbol{\beta}_{n,1}, \cdots, \boldsymbol{\beta}_{n,n}] \in \mathcal{R}^{1\times(n+1)m},$$

then we have $v = 1$ and

$$\mathbf{TA} - \mathbf{GT} = \mathbf{lc} \implies \mathbf{TAT}^{-1} = \mathbf{G} + \mathbf{lcT}^{-1} = \mathbf{G} + \mathbf{lw}$$

Based on the defined $\mathbf{G}$ and $\mathbf{w}$, it is straightforward to derive that

$$\mathbf{z}_d(k+1) = \mathbf{A}_z\mathbf{z}_d(k) + \mathbf{B}_z\mathbf{y}(k) \tag{4.39}$$

$$\hat{\theta}(k) = \mathbf{c}_z\mathbf{z}_d(k) + \mathbf{d}_z\mathbf{y}(k). \tag{4.40}$$

is in the observability canonical form where $\mathbf{A}_z := \mathbf{G} + \mathbf{lw}, \mathbf{B}_z := (\mathbf{H} + \mathbf{l}\boldsymbol{\beta}_{n,n}), \mathbf{c}_z = \mathbf{w}$ and $\mathbf{d}_z = \boldsymbol{\beta}_{n,n}$. It is clear that $\hat{\theta}$ can be considered as a "soft" sensor for the KPI.

For monitoring, assume that the process variables are dynamically changing but the KPI is quasi-steady and denote its reference value as $\bar{\theta}(k)$, which can be *e.g.* obtained from the SCADA system. During the steady KPI periods we have $E(\hat{\theta}(k) - \bar{\theta}(k)) = 0$ and the variance of the estimated KPI is $\hat{\sigma}^2 = E((\hat{\theta}(k) - \bar{\theta}(k))^2)$. For offline training, variance of the estimated KPI can be obtained as

$$\hat{\sigma}^2 = \sum_{k=1}^{N}(\hat{\theta}(k) - \bar{\theta}(k))^2/(N-1).$$

For online monitoring, the T^2 test statistic can be applied as

$$T^2 = \frac{(\hat{\theta}(k) - \bar{\theta}(k))^2}{\hat{\sigma}^2}$$

The corresponding threshold can be determined from the χ^2-distribution as $J_{th,T^2} = \chi^2_{1-\alpha}(1)$.

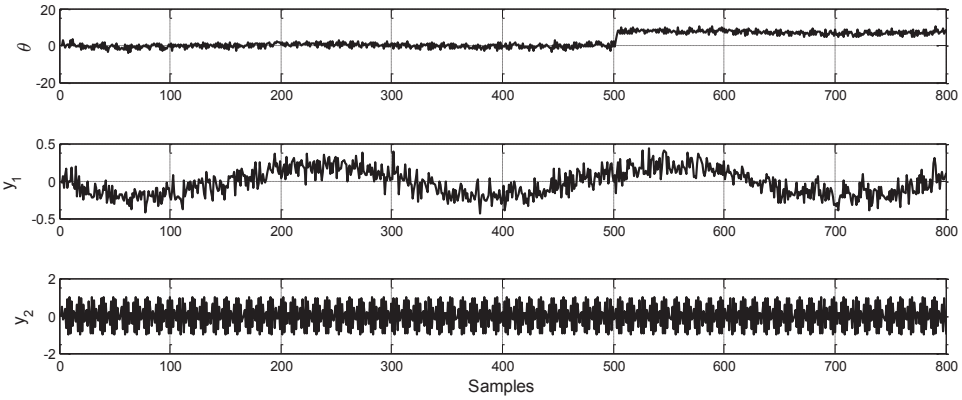
4.5 Numerical examples

In this section, the following dynamic system is considered:

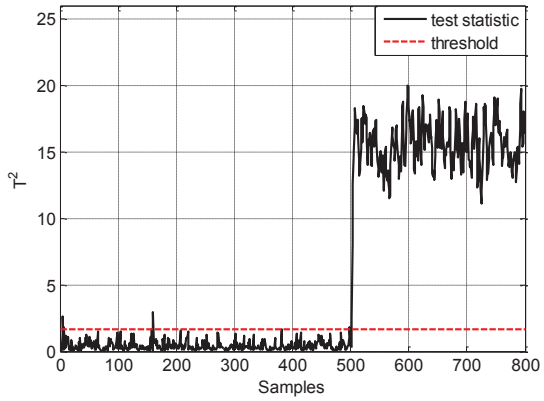
$$\mathbf{z}(k+1) = \mathbf{Az}(k) + \mathbf{By}(k) + \boldsymbol{\eta}(k),$$
$$\theta(k) = \mathbf{cz}(k) + \xi(k)$$

where

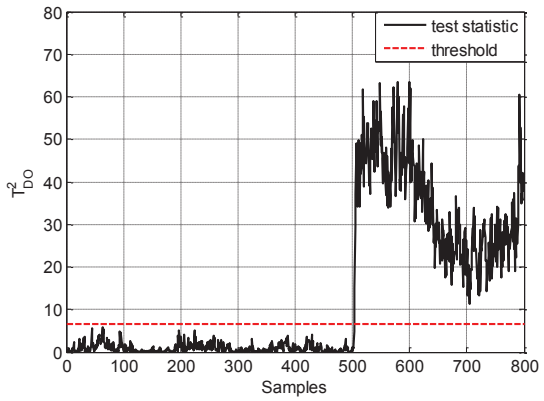
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -0.04 & -0.4 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix},\ \mathbf{z}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ c = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$



(a) Measurements of KPIs and process variables. At the 501st sample, a fault happens in the process. It is not caused by low-level process variable changes and influences the KPI



(b) Monitoring result with parity space based approach



(c) Monitoring result with diagnostic observer

Figure 4.1: Illustration of the kernel representation based process monitoring

$\boldsymbol{\eta}$ is multivariate normally distributed with a zero-mean vector and covariance matrix $diag(0.1, 0.1, 0.1)$, ξ is normally distributed with zero mean and variance 0.01, and η and ξ are mutually uncorrelated. Four hundred samples of fault-free data are generated for the offline training. A fault affecting the KPI has been introduced at the 501st sample. As shown in Figure 4.1-a, both process variables are not affected.

To identify the kernel matrices, we select $s = 3$ and $s_p = 10$. By checking the magnitudes of $\begin{bmatrix} \boldsymbol{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_2 \end{bmatrix}$ which is given in (4.23), n is determined to be 4. All parity vectors are used to build the PS-based residual generator. For constructing the DO, one of those parity vectors is utilized. Figures 4.1-b and 4.1-c show the monitoring results with the PS-based approach and the DO, respectively. It can be observed that both charts have successfully detected the fault. Since the recursive predictor uses only online process

variables (without online KPI data), this type of faults cannot be detected.

4.6 Concluding remarks

This chapter discusses some new results for data-driven design of model-based monitoring techniques for dynamic processes. Different from the standard two-step approach which relies on the system identification, the new methods are able to directly identify the process monitoring systems. Among them, PS-based method is the simplest and is directly built from the kernel matrices. However, it relies on the past data during online phase, which not only consumes valuable computation resources, but also can cause long detection delays. A DO can be easily constructed from any parity vector. It does not need the past data for residual generation and is computationally more efficient. In addition, for the case that the KPI is not online measurable, the recursive predictor can be used. Since it uses only the low-level process measurements, the monitoring performance is restricted. For instance, those faults which do not impact the KPI cannot be detected. Except for the cases discussed in Chapters 3 and 4, many real processes cannot be described by either static equations or ODEs. To describe increasingly complex systems, PDEs are frequently used. The next chapter will focus on the monitoring system design for such processes.

5 KPI monitoring techniques for distributed-parameter processes

Except for static processes and LPPs, DPPs are frequently encountered in the industry as well, especially in continuous manufacturing processes like paper machine or hot strip mill [19, 20, 28, 45, 95]. For these processes, the KPIs, *e.g.* moisture or thickness of the paper, thickness or temperature of the steel strip, are closely related to the enterprise profit and should be automatically monitored. However, the KPI monitoring techniques developed for static processes or LPPs cannot be applied. Since the manipulated variables and the state variables are functions of both the time and the space, their dynamics is generally described by PDEs that are often derived from the fundamental balance of mass, momentum and energy. Traditional approaches for handling DPPs are based on the simplifying assumption that the distributed functions are spatially uniform. However, with ever increasing global competence, it is difficult to meet strict economic and environmental requirements with those traditional techniques.

Motivated by these observations, in the present work, a statistical KPI monitoring framework is proposed for multiple-input and multiple-output (MIMO) DPPs. For the reason of simplicity and considering the fact that most differential equations arising in science and engineering are first or second order [39], we will develop KPI monitoring techniques based on a general description of second order PDEs. The major objective of this chapter is to obtain a *kernel representation* of the MIMO DPPs for residual generation. For this purpose, the concept of *projection* in infinite dimensional space plays an essential role. Aiming at reducing the application effort, data-driven realization of the obtained kernel representation is another objective of this chapter.

5.1 Problem formulation

We consider the DPPs described by the following equations

$$\mathbf{A}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial t^2}+\mathbf{B}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial t\partial x}+\mathbf{C}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial x^2}+\mathbf{D}(x)\frac{\partial\mathbf{z}(x,t)}{\partial t}$$
$$+\mathbf{E}(x)\frac{\partial\mathbf{z}(x,t)}{\partial x}+\mathbf{F}(x)\mathbf{z}(x,t)+\mathbf{G}(x)\mathbf{y}(x,t)+\boldsymbol{\eta}(x,t)=0, \tag{5.1}$$

$$\boldsymbol{\theta}(x,t)=\int_\alpha^\beta\mathbf{H}(x,x')\mathbf{z}(x',t)\mathrm{d}x'+\int_\alpha^\beta\mathbf{I}(x,x')\frac{\partial\mathbf{z}(x',t)}{\partial t}\mathrm{d}x'+\boldsymbol{\xi}(x,t), \tag{5.2}$$
$$\alpha\le x\le\beta,\ t>=0$$

which are subject to either the Dirichlet boundary conditions

$$\mathbf{z}(\alpha,t)=\mathbf{z}_\alpha(t),\ \mathbf{z}(\beta,t)=\mathbf{z}_\beta(t) \tag{5.3}$$

or the Neumann boundary conditions

$$\frac{\partial^j\mathbf{z}(x,t)}{\partial x^j}|_{x=\alpha_j}=\mathbf{z}_{\alpha_j}(t),\ \frac{\partial^j\mathbf{z}(x,t)}{\partial x^j}|_{x=\beta_j}=\mathbf{z}_{\beta_j}(t), j=1,2 \tag{5.4}$$

(or their combinations) and the initial condition

$$\mathbf{z}(x,t)=\mathbf{z}_0(x) \tag{5.5}$$

where $\mathbf{z}(x,t)=[z_1(x,t)\cdots z_n(x,t)]^T\in\mathcal{H}^n$ denotes a vector of the state variables in the Hilbert space, $\mathbf{y}(x,t)=[y_1(x,t)\cdots y_m(x,t)]^T\in\mathcal{H}^m$ and $\boldsymbol{\theta}(x,t)=[\theta_1(x,t)\cdots\theta_l(x,t)]^T\in\mathcal{H}^l$ denote the manipulated low-level process variables and the measured KPIs[1], receptively (they are called "snapshots" of the corresponding variables), $\boldsymbol{\eta}(x,t)\in\mathcal{H}^n$ and $\boldsymbol{\xi}(x,t)\in\mathcal{H}^m$ represent random process and measurement noises, $\mathbf{A}(x),\mathbf{B}(x),\mathbf{C}(x),\mathbf{D}(x),\mathbf{E}(x),\mathbf{F}(x),\mathbf{G}(x),\mathbf{H}(x,x')$ and $\mathbf{I}(x,x')$ are matrices of functions with approximate dimensions, $x\in[\alpha,\beta]$ denotes the spatial coordinate and t denotes the time.

The objective of this chapter is to establish a statistical KPI monitoring framework for practical DPPs. To achieve it, the following tasks are formulated:

- establishing a kernel representation of (5.1) in the discrete-time form as

$$\mathbf{r}(x')=\int_\alpha^\beta[\boldsymbol{\Psi}^\perp(p,x,x')]\begin{bmatrix}\mathbf{y}(x)\\\boldsymbol{\theta}(x)\end{bmatrix}\mathrm{d}x \tag{5.6}$$

  where the effects of the manipulated variables are eliminated. Here $\mathbf{y}(x)$ and $\boldsymbol{\theta}(x)$ are the z-transformations of $\mathbf{y}(x,t)$ and $\boldsymbol{\theta}(x,t)$ with respect to t, p denotes the z-transformation operator.

[1](5.2) is a general form of sensor measurement functions. Currently, most KPIs are measured at several fixed locations. However, with the rapid developing sensing technologies, they can also be measured over the whole spatial range, *e.g.* using high temperature infrared cameras (http://www.directindustry.com), the temperature profile of the hot strip can be measured.

- designing a residual evaluator by
  - utilizing a proper GLR test function and
  - determining the threshold J_{th} in the statistical framework with deterministic projection error
- realizing the kernel representation (5.6) based on the process I/O data and the equivalent observer-based implementation form.

5.2 The concept of projection in infinite dimensional case

The concept of *projection* (of functions) in infinite dimensional case plays an essential role for designing a kernel representation based residual generator for (5.1). To understand it, let us first review the *orthogonal projection* of vectors on a known finite dimensional subspace.

Given an arbitrary vector $\mathbf{q} \in \mathcal{R}^n$ and a full column-rank matrix $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_\gamma \end{bmatrix} \in \mathcal{R}^{n\times\gamma}$, we want to find an optimal $\hat{\mathbf{q}} \in \mathcal{R}^n$ that lies in the column space of $\mathbf{P}$ satisfying

$$(\mathbf{q}-\hat{\mathbf{q}})^T(\mathbf{q}-\hat{\mathbf{q}}) = \min_{\mathbf{r}\in\mathbf{P}}(\mathbf{q}-\mathbf{r})^T(\mathbf{q}-\mathbf{r}). \tag{5.7}$$

Let us denote $\hat{\mathbf{q}} = \sum_{i=1}^{\gamma} z_i\mathbf{p}_i = \mathbf{P}\mathbf{z}$, where $\mathbf{z} = [z_1, \cdots, z_\gamma]^T, z_i \in \mathcal{R}, i = 1, \cdots, \gamma$. The task for projection is actually to find the unknown coefficients $z_i, i = 1, \cdots, \gamma$. According to Theorem B.1, we know that the optimal $\mathbf{q}$ can be found by solving

$$(\mathbf{q}-\hat{\mathbf{q}})^T\mathbf{p}_i = 0, \text{ for } i = 1, \cdots, \gamma. \tag{5.8}$$

Substitute $\hat{\mathbf{q}} = \mathbf{P}\mathbf{z}$ into (5.8) we can obtain

$$\mathbf{z} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{q} \Rightarrow \hat{\mathbf{q}} = \mathbf{P}\mathbf{z} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{q} := \text{proj}_{\mathbf{P}}\,\mathbf{q}$$

where $\hat{\mathbf{q}}$ is the orthogonal projection of $\mathbf{q}$ on the subspace $\mathbf{P}$.

Example 5.1. *As illustrated in Figure 5.1, let* $\mathbf{q} = [2, 1]^T$ *("raw data") be an arbitrary vector in* $\mathcal{R}^2$ *and* $\mathbf{p}_1 = [3, 1]^T$ *be a given subspace/direction in* $\mathcal{R}^2$*. Based on the above formula we can get* $\hat{\mathbf{q}} = [2.1, 0.7]^T$ *("projected data"). It can easily be verified that* $\hat{\mathbf{q}} - \mathbf{q}$ *is perpendicular to* $\mathbf{p}_1$ *thus the best estimate of* $\mathbf{q}$ *using* $\mathbf{p}_1$ *is* $\hat{\mathbf{q}}$*.*

Differently, in the infinite dimensional case the basis of subspace $\mathbf{P}$ as well as the arbitrary function $\mathbf{q}$ are in the Hilbert space [39]. Let us denote $\mathbf{p}(x) = [p_1(x), \cdots, p_\gamma(x)]^T, p_i \in \mathcal{H}, i = 1, \cdots, \gamma, q \in \mathcal{H}$. Note that here q is no longer some

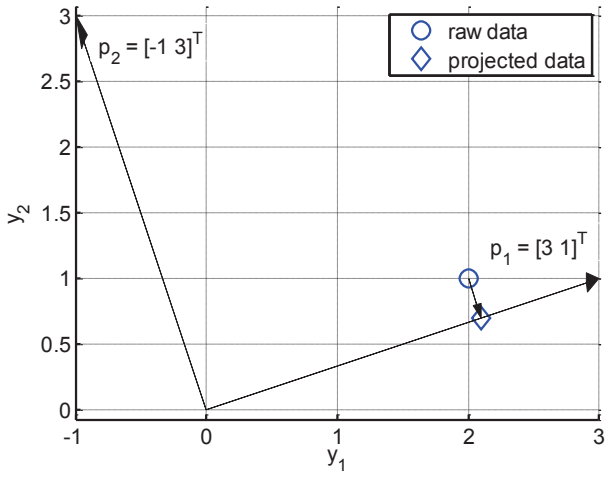


Figure 5.1: Orthogonal projection of a vector in finite dimensional space

constant (vector), but a function of x. The task is to approximate $q(x)$ in the subspace with $\mathbf{p}(x)$ as basis functions. The optimal approximation $\hat{q}(x)$ should satisfy

$$\int_\alpha^\beta \left(q(x) - \hat{q}(x)\right)^2 \mathrm{d}x = \min_{r(x) \in span\{\mathbf{p}(x)\}} \int_\alpha^\beta \left(q(x) - r(x)\right)^2 \mathrm{d}x. \tag{5.9}$$

According to Theorem B.2, the optimal $\hat{q}(x)$ can be found by solving

$$\int_\alpha^\beta \left(q(x) - \hat{q}(x)\right) p_i(x) \mathrm{d}x = 0 \text{ for } i = 1, \cdots, \gamma. \tag{5.10}$$

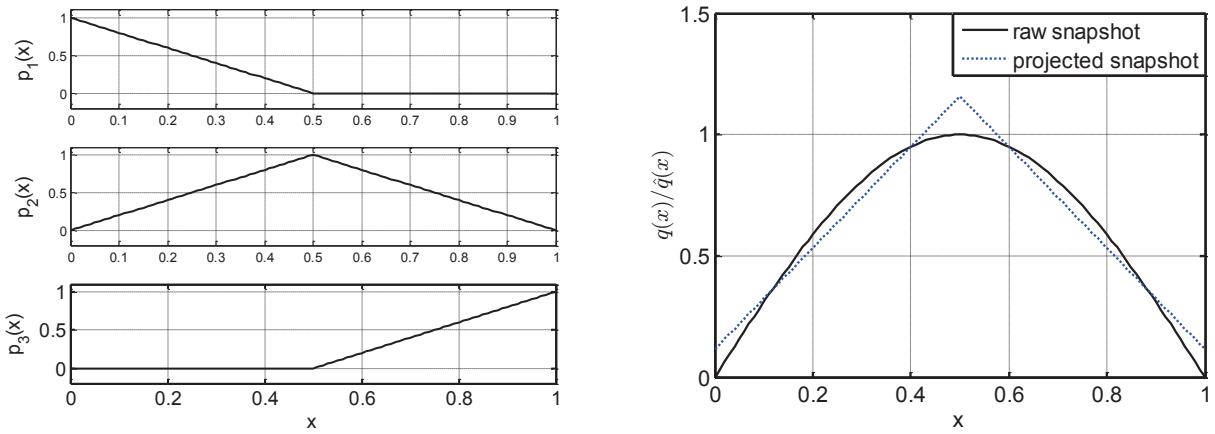
By substituting $\hat{q}(x) = \sum_{i=1}^{\gamma} z_i p_i(x)$ into (5.10) we have

$$\begin{bmatrix} z_1 \\ \vdots \\ z_\gamma \end{bmatrix} = \begin{bmatrix} \int_\alpha^\beta p_1(x)p_1(x)\mathrm{d}x & \cdots & \int_\alpha^\beta p_1(x)p_\gamma(x)\mathrm{d}x \\ \vdots & \ddots & \vdots \\ \int_\alpha^\beta p_\gamma(x)p_1(x)\mathrm{d}x & \cdots & \int_\alpha^\beta p_\gamma(x)p_\gamma(x)\mathrm{d}x \end{bmatrix}^{-1} \begin{bmatrix} \int_\alpha^\beta p_1(x)q(x)\mathrm{d}x \\ \vdots \\ \int_\alpha^\beta p_\gamma(x)q(x)\mathrm{d}x \end{bmatrix} \in \mathcal{R}^\gamma, \tag{5.11}$$

$$\Rightarrow z := \left(\int_\alpha^\beta \mathbf{p}(x)\mathbf{p}^T(x)\mathrm{d}x\right)^{-1} \int_\alpha^\beta \mathbf{p}(x)q(x)\mathrm{d}x. \tag{5.12}$$

Thus the projection of $q(x)$ onto $\mathbf{p}(x)$ is given as $\hat{q}(x) = \mathbf{p}^T(x)\mathbf{z} := \mathrm{proj}_{\mathbf{p}(x)}\, q(x)$ where $\mathbf{z} = [z_1, \cdots, z_\gamma]^T \in \mathcal{R}^\gamma$.

Example 5.2. *As depicted in Figure 5.2, $q(x) = sin(\pi x)$ is an arbitrary function in $\mathcal{H}$ ("raw snapshot") and $p_i(x), i = 1, 2, 3$, is the given basis function. Using the above formula we can get $\mathbf{z} = [0.1148, 1.1584, 0.1148]^T$. Based on it, $\hat{q} = \sum_{i=1}^{3} z(i)p_i(x)$ is calculated and is plotted in Figure 5.2-b ("projected snapshot"). $\hat{q}(x)$ is the best approximation of $q(x)$ in span$\{\mathbf{p}(x)\}$ with the error term $\int_0^1 (q(x) - \hat{q}(x))^2 \mathrm{d}x$ minimized.*



(a) Three defined basis functions (b) Projection of raw snapshot

Figure 5.2: Projection of a function in infinite dimensional space

5.3 Design of KPI monitoring systems for DPPs

5.3.1 Projection-based process lumping

From (5.1) we know that $\mathbf{A}(x), \cdots, \mathbf{F}(x)$ are model parameter matrices, $\mathbf{y}(x,t)$ is a given manipulated vector and $z_i(x,t)$, for $i = 1, \cdots, n$, is the i-th infinite dimensional state variable. In order to design an implementable monitoring system, the order of state variables must be reduced to a finite one. For this purpose, the concept of projection discussed in the previous section plays a central role and forms the basis for an optimal solution. To simplify notations, let us define the following operations [39]:

$$(m(x), n(x)) = \int_\alpha^\beta m(x)n(x)\mathrm{d}x, \ ||m(x)||^2 = \int_\alpha^\beta m^2(x)\mathrm{d}x \tag{5.13}$$

where $m(x), n(x) \in \mathcal{H}$.

In addition, a linear operator[2] on $\mathbf{z}(x,t)$ is defined as

$$\begin{aligned}\mathcal{L}(\mathbf{z}(x,t)) = \mathbf{A}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial t^2} + \mathbf{B}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial t\partial x} + \mathbf{C}(x)\frac{\partial^2\mathbf{z}(x,t)}{\partial x^2} \\ +\mathbf{D}(x)\frac{\partial\mathbf{z}(x,t)}{\partial t} + \mathbf{E}(x)\frac{\partial\mathbf{z}(x,t)}{\partial x} + \mathbf{F}(x)\mathbf{z}(x,t), x \in [\alpha,\beta].\end{aligned} \tag{5.14}$$

Based on a given subspace, we want to represent $\mathbf{z}(x,t)$ in such a way that the total approximation error is minimized, *i.e.*

$$tr\left(||\mathcal{L}(\mathbf{z}(x,t)) - \mathcal{L}(\hat{\mathbf{z}}(x,t))||^2\right) = \min. \tag{5.15}$$

[2]An operator is a function with functions as inputs. It is equivalent to the first "function".

where

$$||\mathcal{L}(\mathbf{z}(x,t)) - \mathcal{L}(\hat{\mathbf{z}}(x,t))||^2 := \begin{bmatrix} ||\mathcal{L}_1(\mathbf{z}(x,t)) - \mathcal{L}_1(\hat{\mathbf{z}}(x,t))||^2 & \cdots & (\mathcal{L}_1(\mathbf{z}(x,t)) - \mathcal{L}_1(\hat{\mathbf{z}}(x,t)), \mathcal{L}_n(\mathbf{z}(x,t)) - \mathcal{L}_n(\hat{\mathbf{z}}(x,t))) \\ \vdots & \ddots & \vdots \\ (\mathcal{L}_n(\mathbf{z}(x,t)) - \mathcal{L}_n(\hat{\mathbf{z}}(x,t)), \mathcal{L}_1(\mathbf{z}(x,t)) - \mathcal{L}_1(\hat{\mathbf{z}}(x,t))) & \cdots & ||\mathcal{L}_n(\mathbf{z}(x,t)) - \mathcal{L}_n(\hat{\mathbf{z}}(x,t))||^2 \end{bmatrix}$$

with

$$\mathcal{L}(\cdot) := \begin{bmatrix} \mathcal{L}_1(\cdot) \\ \vdots \\ \mathcal{L}_n(\cdot) \end{bmatrix} \in \mathcal{H}^n.$$

In the following, we will apply the concept of projection to solve this optimization problem. Let $\mathcal{V}(x) := span\{v_1(x), \cdots, v_\gamma(x), v_{\gamma+1}(x), \cdots\}$ denotes the infinite dimensional function space that $z_i(x,t), i = 1, \cdots, n$, belongs to. Equivalently, we can write

$$z_i(x,t) = \sum_{j=1}^{\infty} \bar{z}^*_{j,i}(t) v_j(x), i = 1, \cdots, n \tag{5.16}$$

where $\bar{z}^*_{j,i}(t),$ for $i = 1, \cdots, n,\ j = 1, \cdots, \infty$, is the weighting coefficient.

Note that the dimension of $\mathcal{V}(x)$ is infinite. In order to reduce the order of the problem, a subspace $\hat{\mathcal{V}}(x) := span\{v_1(x), \cdots, v_\gamma(x)\} \subset \mathcal{V}(x)$ is defined *a priori*. We want to find the best representation of $\mathbf{z}(x,t)$ from $\hat{\mathcal{V}}(x)$ in the form of

$$\hat{z}_i(x,t) = \sum_{j=1}^{\gamma} \bar{z}_{j,i}(t) v_j(x), i = 1, \cdots, n \tag{5.17}$$

such that (5.15) holds. Here $\bar{z}_{j,i},$ for $i = 1, \cdots, n,\ j = 1, \cdots, \gamma$, is the weighting coefficient that needs to be determined. By applying the concept of projection in infinite dimensional case, we have the Theorem B.3. Based on it, the solution of (5.15) can be formulated as

$$(\mathcal{L}(\mathbf{z}(x,t) - \hat{\mathbf{z}}(x,t)), v_i(x)) = 0, \forall\ i = 1, \cdots, \gamma. \tag{5.18}$$

$\hat{\mathbf{z}}(x,t)$ satisfying (5.18) is defined as the projection:

$$\hat{\mathbf{z}}(x,t) = \text{proj}_{\hat{\mathcal{V}}|\mathcal{L}}\, \mathbf{z}(x,t). \tag{5.19}$$

which gives the best estimate of $\mathbf{z}(x,t)$ from $\hat{\mathcal{V}}(x)$ for the $\mathcal{L}$-norm defined as

$$||\cdot||^2_{\mathcal{L}} = tr\left(||\mathcal{L}(\cdot)||^2\right).$$

Different from the standard projection discussed in the previous section, the coefficient $\bar{z}_{j,i},\ i = 1, \cdots, i,\ j = 1, \cdots, \gamma$, from (5.17) cannot be obtained directly. To this end, the original PDE system (5.1) needs to be reformulated into its *weak form* or *variational form* as

$$\Big(\mathcal{L}(\mathbf{z}(x,t)) + \mathbf{G}(x)\mathbf{y}(x,t) + \boldsymbol{\eta}(x,t), v(x)\Big) :=$$
$$\int_{\alpha}^{\beta} \Big(\mathcal{L}(\mathbf{z}(x,t)) + \mathbf{G}(x)\mathbf{y}(x,t) + \boldsymbol{\eta}(x,t)\Big)v(x)\mathrm{d}x = 0, \forall\ v(x) \in \mathcal{V}(x) \tag{5.20}$$

Note that the above equation needs to be held for any function in $\mathcal{V}(x)$ and it results in infinite equations. Using the given subspace $\hat{\mathcal{V}}(x)$, we can approximate the weak form by solving

$$\Big(\Big(\mathcal{L}(\hat{\mathbf{z}}(x,t)) + \mathbf{G}(x)y(x,t) + \boldsymbol{\eta}(x,t)\Big), v_i(x)\Big) = 0, \forall\ i = 1, \cdots, \gamma, \tag{5.21}$$

which is an ODE system with a minimal number of $n\gamma$ equations.

Subtracting (5.21) from (5.20), we see that

$$(\mathcal{L}(\mathbf{z}(x,t)) - \mathcal{L}(\hat{\mathbf{z}}(x,t)), v_i(x)) = 0, \forall\ i = 1, \cdots, \gamma.$$

Based on Theorem B.3, we know that the $\hat{\mathbf{z}}(x,t)$ given by (5.21) provides the best approximation of $\mathbf{z}(x,t)$ in the sense of $\mathcal{L}$-norm, *i.e.* (5.15) is satisfied.

Remark: The solution given by (5.21) is known in mathematics as the *Galerkin approximation* of the true solution $\mathbf{z}(x,t)$. It serves as the core of finite element method. Basics about numerical approaches for PDE can be found *e.g.* in [39] and references therein.

Before giving the final approximation solution, let us define the following notations

$$\boldsymbol{\Phi}(x) = \begin{bmatrix} \phi_{11}(x) & \cdots & \phi_{1n}(x) \\ \vdots & \ddots & \vdots \\ \phi_{n1}(x) & \cdots & \phi_{nn}(x) \end{bmatrix} \in \mathcal{H}^{n\times n} \tag{5.22}$$

$$\bar{\boldsymbol{\Phi}}_{ij} = \begin{bmatrix} (v_1(x), \phi_{ij}(x)v_1(x)) & \cdots & (v_1(x), \phi_{ij}(x)v_\gamma(x)) \\ \vdots & \ddots & \vdots \\ (v_\gamma(x), \phi_{ij}(x)v_1(x)) & \cdots & (v_\gamma(x), \phi_{ij}(x)v_\gamma(x)) \end{bmatrix} \in \mathcal{R}^{\gamma\times\gamma}, \tag{5.23}$$

$$\bar{\boldsymbol{\Phi}} = \begin{bmatrix} \bar{\boldsymbol{\Phi}}_{11} & \cdots & \bar{\boldsymbol{\Phi}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\boldsymbol{\Phi}}_{n1} & \cdots & \bar{\boldsymbol{\Phi}}_{nn} \end{bmatrix} \in \mathcal{R}^{n\gamma\times n\gamma} \tag{5.24}$$

where $\hat{\mathcal{V}} = span\{v_1(x), \cdots, v_\gamma(x)\}$ is the given subspace. The lumped model (5.21) can be reformulated as

$$\bar{\mathbf{A}}\ddot{\bar{\mathbf{z}}} + \bar{\mathbf{B}}\dot{\bar{\mathbf{z}}} + \bar{\mathbf{C}}\bar{\mathbf{z}} + \bar{\mathbf{D}}\dot{\bar{\mathbf{z}}} + \bar{\mathbf{E}}\bar{\mathbf{z}} + \bar{\mathbf{F}}\bar{\mathbf{z}} + \bar{\mathbf{G}}\bar{\mathbf{y}} + \bar{\boldsymbol{\eta}} = 0 \tag{5.25}$$

where $\bar{\mathbf{A}}, \bar{\mathbf{D}}, \bar{\mathbf{E}}, \bar{\mathbf{F}} \in \mathcal{R}^{n\gamma\times n\gamma}$ are constructed from $\mathbf{A}(x), \mathbf{D}(x), \mathbf{E}(x), \mathbf{F}(x) \in \mathcal{H}^{n\times n}$ respectively in the same way as $\bar{\boldsymbol{\Phi}}$ from $\boldsymbol{\Phi}(x)$, $\bar{\mathbf{B}} \in \mathcal{R}^{n\gamma\times n\gamma}$ is constructed from $\mathbf{B}(x) \in \mathcal{H}^{n\times n}$ similar to (5.22-5.24) except that (5.23) is replaced by

$$\bar{\mathbf{B}}_{ij} = \begin{bmatrix} (v_1(x), b_{ij}(x)\frac{\mathrm{d}v_1}{\mathrm{d}x}(x)) & \cdots & (v_1(x), b_{ij}(x)\frac{\mathrm{d}v_\gamma}{\mathrm{d}x}(x)) \\ \vdots & \ddots & \vdots \\ (v_\gamma(x), b_{ij}(x)\frac{\mathrm{d}v_1}{\mathrm{d}x}(x)) & \cdots & (v_\gamma(x), b_{ij}(x)\frac{\mathrm{d}v_\gamma}{\mathrm{d}x}(x)) \end{bmatrix} \in \mathcal{R}^{\gamma\times\gamma}, \tag{5.26}$$

$\bar{\mathbf{C}} \in \mathcal{R}^{n\gamma\times n\gamma}$ is from $\mathbf{C}(x) \in \mathcal{H}^{n\times n}$ with (5.23) replaced by

$$\begin{aligned} \bar{\mathbf{C}}_{ij} = &\begin{bmatrix} c_{ij}(x)v_1(x)\frac{\mathrm{d}v_1(x)}{\mathrm{d}x}|_\alpha^\beta & \cdots & c_{ij}(x)v_1(x)\frac{\mathrm{d}v_\gamma(x)}{\mathrm{d}x}|_\alpha^\beta \\ \vdots & \ddots & \vdots \\ c_{ij}(x)v_\gamma(x)\frac{\mathrm{d}v_1(x)}{\mathrm{d}x}|_\alpha^\beta & \cdots & c_{ij}(x)v_\gamma(x)\frac{\mathrm{d}v_\gamma(x)}{\mathrm{d}x}|_\alpha^\beta \end{bmatrix} \\ &- \begin{bmatrix} (\frac{d}{\mathrm{d}x}(c_{ij}(x)v_1(x)), \frac{\mathrm{d}v_1(x)}{\mathrm{d}x}) & \cdots & (\frac{d}{\mathrm{d}x}(c_{ij}(x)v_1(x)), \frac{\mathrm{d}v_\gamma(x)}{\mathrm{d}x}) \\ \vdots & \ddots & \vdots \\ (\frac{d}{\mathrm{d}x}(c_{ij}(x)v_\gamma(x)), \frac{\mathrm{d}v_1(x)}{\mathrm{d}x}) & \cdots & (\frac{d}{\mathrm{d}x}(c_{ij}(x)v_\gamma(x)), \frac{\mathrm{d}v_\gamma(x)}{\mathrm{d}x}) \end{bmatrix}, \end{aligned} \tag{5.27}$$

$\bar{\mathbf{G}} \in \mathcal{R}^{n\gamma\times m\gamma}$ is from $\mathbf{G}(x) \in \mathcal{H}^{n\times m}$ with (5.23) constructed as

$$\bar{\mathbf{G}}_{ij} = \begin{bmatrix} (v_1(x), g_{ij}(x)v_1(x)) & \cdots & (v_1(x), g_{ij}(x)v_\gamma(x)) \\ \vdots & \ddots & \vdots \\ (v_\gamma(x), g_{ij}(x)v_1(x)) & \cdots & (v_\gamma(x), g_{ij}(x)v_\gamma(x)) \end{bmatrix} \in \mathcal{R}^{\gamma\times\gamma}, \tag{5.28}$$

the lumped state vector $\bar{\mathbf{z}}$ is defined as

$$\bar{\mathbf{z}} = \mathrm{vec}\left(\begin{bmatrix} \bar{z}_{1,1} & \cdots & \bar{z}_{1,n} \\ \vdots & \ddots & \vdots \\ \bar{z}_{\gamma,1} & \cdots & \bar{z}_{\gamma,n} \end{bmatrix}\right) \in \mathcal{R}^{n\gamma\times 1} \tag{5.29}$$

and the lumped process variable $\bar{\mathbf{y}}$ is

$$\bar{\mathbf{y}} := (\hat{\mathbf{v}}(x), \mathbf{y}(x,t)) = \mathrm{vec}\left(\begin{bmatrix} (v_1(x), y_1(x,t)) & \cdots & (v_1(x), y_m(x,t)) \\ \vdots & \ddots & \vdots \\ (v_\gamma(x), y_1(x,t)) & \cdots & (v_\gamma(x), y_m(x,t)) \end{bmatrix}\right) \in \mathcal{R}^{m\gamma\times 1} \tag{5.30}$$

and

$$\bar{\boldsymbol{\eta}} := (\hat{\mathbf{v}}(x), \boldsymbol{\eta}(x,t)) = \mathrm{vec}\left(\begin{bmatrix} (v_1(x), \eta_1(x,t)) & \cdots & (v_1(x), \eta_n(x,t)) \\ \vdots & \ddots & \vdots \\ (v_\gamma(x), \eta_1(x,t)) & \cdots & (v_\gamma(x), \eta_n(x,t)) \end{bmatrix}\right) \in \mathcal{R}^{n\gamma \times 1} \quad (5.31)$$

with

$$\hat{\mathbf{v}}(x) = \begin{bmatrix} v_1(x) & \cdots & v_\gamma(x) \end{bmatrix}^T \in \mathcal{H}^\gamma.$$

Finally, the second order system (5.25) can be arranged as a first order one as

$$\dot{\mathbf{z}} = \mathbf{A}_c \mathbf{z} + \mathbf{B}_c \bar{\mathbf{y}} + \mathbf{E}_c \bar{\boldsymbol{\eta}} \quad (5.32)$$

where

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\bar{\mathbf{A}}^{-1}(\bar{\mathbf{C}} + \bar{\mathbf{E}} + \bar{\mathbf{F}}) & -\bar{\mathbf{A}}^{-1}(\bar{\mathbf{B}} + \bar{\mathbf{D}}) \end{bmatrix} \in \mathcal{R}^{2n\gamma \times 2n\gamma},$$

$$\mathbf{B}_c = \begin{bmatrix} \mathbf{0} \\ -\bar{\mathbf{A}}^{-1}\bar{\mathbf{G}} \end{bmatrix} \in \mathcal{R}^{2n\gamma \times m\gamma},\ \mathbf{E}_c = \begin{bmatrix} \mathbf{0} \\ -\bar{\mathbf{A}}^{-1} \end{bmatrix} \in \mathcal{R}^{2n\gamma \times n\gamma}\ \mathbf{z} = \begin{bmatrix} \bar{\mathbf{z}} \\ \dot{\bar{\mathbf{z}}} \end{bmatrix} \in \mathcal{R}^{2n\gamma}.$$

Remark: Note that $\mathbf{z}(t)$ or $\mathbf{z}(k)$ is used to denote the lumped state vector for both continuous- or discrete-time descriptions. For the simplicity of notations, we drop t or k in the equations if it is not critical. It is different from $\mathbf{z}(x,t)$ which is dependent on the space.

Similarly, the output equation (5.2) can be represented as

$$\boldsymbol{\theta}(x,t) = \mathbf{C}_c(x)\mathbf{z}(t) + \boldsymbol{\xi}(x,t) \quad (5.33)$$

where

$$\mathbf{C}_c(x) = \begin{bmatrix} \mathbf{C}_{c1}(x) & \mathbf{C}_{c2}(x) \end{bmatrix},\ \mathbf{C}_{c1}(x) = \begin{bmatrix} \mathbf{C}_{c1,1,1}(x) & \cdots & \mathbf{C}_{c1,1,n}(x) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{c1,l,1}(x) & \cdots & \mathbf{C}_{c1,l,n}(x) \end{bmatrix} \in \mathcal{H}^{l \times n\gamma},$$

$$\mathbf{C}_{c2}(x) = \begin{bmatrix} \mathbf{C}_{c2,1,1}(x) & \cdots & \mathbf{C}_{c2,1,n}(x) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{c2,l,1}(x) & \cdots & \mathbf{C}_{c2,l,n}(x) \end{bmatrix} \in \mathcal{H}^{l \times n\gamma},$$

$$\mathbf{C}_{c1,i,j}(x) = \begin{bmatrix} (h_{ij}(x,x'), v_1(x')) & \cdots & (h_{ij}(x,x'), v_\gamma(x')) \end{bmatrix} \in \mathcal{H}^{1 \times \gamma},$$

$$\mathbf{C}_{c2,i,j}(x) = \begin{bmatrix} (i_{ij}(x,x'), v_1(x')) & \cdots & (i_{ij}(x,x'), v_\gamma(x')) \end{bmatrix} \in \mathcal{H}^{1 \times \gamma}.$$

with $h_{ij}(x,x')$ and $i_{ij}(x,x')$ denoting the elements of $\mathbf{H}(x,x')$ and $\mathbf{I}(x,x')$, respectively.

Since our process monitoring algorithms will be implemented in digital computers, it is convenient to rewrite (5.32) and (5.33) into time-discrete form as

$$\mathbf{z}(k+1) = \mathbf{A}_d\mathbf{z}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{E}_d(\hat{\mathbf{v}}(x), \boldsymbol{\eta}(x,k)) \tag{5.34}$$

$$\boldsymbol{\theta}(x,k) = \mathbf{C}_d(x)\mathbf{z}(k) + \boldsymbol{\xi}(x,k), \tag{5.35}$$

where $\mathbf{A}_d, \mathbf{B}_d$ and $\mathbf{E}_d$ are obtained through temporal discretization and $\mathbf{C}_d(x)$ represents the measurement equation.

5.3.2 Kernel representation based residual generation

Based on the lumped description (5.34-5.35), we propose the following observer

$$\hat{\mathbf{z}}(k+1) = \mathbf{A}_d\hat{\mathbf{z}}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \Big(\mathbf{L}(x), \boldsymbol{\theta}(x,k) - \hat{\boldsymbol{\theta}}(x,k)\Big) \tag{5.36}$$

$$\hat{\boldsymbol{\theta}}(x,k) = \mathbf{C}_d(x)\hat{\mathbf{z}}(k),\ \mathbf{r}(x,k) = \boldsymbol{\theta}(x,k) - \hat{\boldsymbol{\theta}}(x,k) \tag{5.37}$$

where

$$\mathbf{L}(x) = \begin{bmatrix} l_{1,1}(x) & \cdots & l_{1,l}(x) \\ \vdots & \ddots & \vdots \\ l_{2n\gamma,1}(x) & \cdots & l_{2n\gamma,l}(x) \end{bmatrix} \in \mathcal{H}^{2n\gamma\times l} \tag{5.38}$$

is an appropriately chosen observer gain matrix in $\mathcal{H}$-space. The error $\mathbf{r}(x,k)$ will serve as the residual signal used for monitoring. By defining $\mathbf{e}(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k)$, the observer dynamics is given as

$$\mathbf{e}(k+1) = (\mathbf{A}_d - (\mathbf{L}(x), \mathbf{C}_d(x))\,\mathbf{e}(k) + \mathbf{E}_d(\hat{\mathbf{v}}(x), \boldsymbol{\eta}(x,k)) - (\mathbf{L}(x), \boldsymbol{\xi}(x,k)) \tag{5.39}$$

$$\mathbf{r}(x,k) = \mathbf{C}_d(x)\mathbf{e}(k) + \boldsymbol{\xi}(x,k) \tag{5.40}$$

where $\mathbf{L}(x)$ should be selected in such a way that $(\mathbf{A}_d - (\mathbf{L}(x), \mathbf{C}_d(x))$ is stable. Since $\mathbf{L}(x)$ is a matrix with elements as functions of x, the standard observer design theory discussed in Chapter 2 cannot be applied. In the following, we will propose a new design scheme for $\mathbf{L}(x)$.

For convenience, let us assume $\mathbf{L}(x)$ can be designed based on $\hat{\mathbf{v}}(x)$ which is defined in the previous section. Denote

$$l_{i,j}(x) = \sum_{k=1}^{\gamma} \alpha_{i,j,k}\ v_k(x),\ i = 1,\cdots,2n\gamma,\ j = 1,\cdots,l, \tag{5.41}$$

then we can derive

$$(\mathbf{L}(x), \mathbf{C}_d(x)) = \mathbf{L}_L \mathbf{C}_L \tag{5.42}$$

where

$$\mathbf{L}_L = \begin{bmatrix} \mathbf{L}_{L,1} & \cdots & \mathbf{L}_{L,l} \end{bmatrix} \in \mathcal{R}^{2n\gamma \times l\gamma}, \mathbf{C}_L = \begin{bmatrix} \mathbf{C}_{L,1} \\ \vdots \\ \mathbf{C}_{L,l} \end{bmatrix} \in \mathcal{R}^{l\gamma \times 2n\gamma}, \mathbf{L}_{L,j} = \begin{bmatrix} \alpha_{1,j,1} & \cdots & \alpha_{1,j,\gamma} \\ \vdots & \ddots & \vdots \\ \alpha_{2n\gamma,j,1} & \cdots & \alpha_{2n\gamma,j,\gamma} \end{bmatrix},$$

$$\mathbf{C}_{L,j} = \begin{bmatrix} (v_1(x), c_{d,j,1}(x)) & \cdots & (v_1(x), c_{d,j,2n\gamma}(x)) \\ \vdots & \ddots & \vdots \\ (v_\gamma(x), c_{d,j,1}(x)) & \cdots & (v_\gamma(x), c_{d,j,2n\gamma}(x)) \end{bmatrix}, j = 1, \cdots, l$$

with

$$\mathbf{C}_d(x) = \begin{bmatrix} c_{d,j,1}(x) & \cdots & c_{d,j,2n\gamma}(x) \\ \vdots & \ddots & \vdots \\ c_{d,j,1}(x) & \cdots & c_{d,j,2n\gamma}(x) \end{bmatrix}.$$

Thus under the assumption that $(\mathbf{A}_d, \mathbf{C}_L)$ is observable, the matrix $\mathbf{L}_L$ can be designed using the standard observer theory, *e.g.* Matlab functions or simply $\mathbf{L}_L = (\mathbf{A}_d - \mathbf{A}_L)\mathbf{C}_L^+$ where $\mathbf{A}_L$ is a desired observer dynamic matrix. Note that $\mathbf{L}_L$ contains the weighting coefficients for (5.41). Based on it, the observer matrix $\mathbf{L}(x)$ can be established as

$$\mathbf{L}(x) = \begin{bmatrix} \mathbf{L}_{L,1}\hat{\mathbf{v}}(x) & \cdots & \mathbf{L}_{L,l}\hat{\mathbf{v}}(x) \end{bmatrix}. \tag{5.43}$$

As a result, the implementation form of (5.36-5.37) is

$$\hat{\mathbf{z}}(k+1) = \bar{\mathbf{A}}_d \hat{\mathbf{z}}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + (\mathbf{L}(x), \boldsymbol{\theta}(x,k)) \tag{5.44}$$

$$\mathbf{r}(x,k) = \boldsymbol{\theta}(x,k) - \mathbf{C}_d(x)\hat{\mathbf{z}}(k) \tag{5.45}$$

where $\bar{\mathbf{A}}_d = (\mathbf{A}_d - (\mathbf{L}(x), \mathbf{C}_d(x)))$. The equivalent *kernel representation* is written as

$$\begin{aligned} \mathbf{r}(x') &= \left(\begin{bmatrix} -\hat{\mathbf{N}}(p,x,x') & \hat{\mathbf{M}}(p,x,x') \end{bmatrix} \begin{bmatrix} \mathbf{y}(x) \\ \boldsymbol{\theta}(x) \end{bmatrix} \right) \\ &= (-\hat{\mathbf{N}}(p,x,x'), \mathbf{y}(x)) + (\hat{\mathbf{M}}(p,x,x'), \boldsymbol{\theta}(x)) \end{aligned} \tag{5.46}$$

where

$$\begin{aligned} \hat{\mathbf{N}}(p,x,x') &= \mathbf{C}_d(x')(p\mathbf{I} - \bar{\mathbf{A}}_d)^{-1}\mathbf{B}_d\hat{\mathbf{v}}(x), \\ \hat{\mathbf{M}}(p,x,x') &= \mathbf{I} - \mathbf{C}_d(x')(p\mathbf{I} - \bar{\mathbf{A}}_d)^{-1}\mathbf{L}(x). \end{aligned}$$

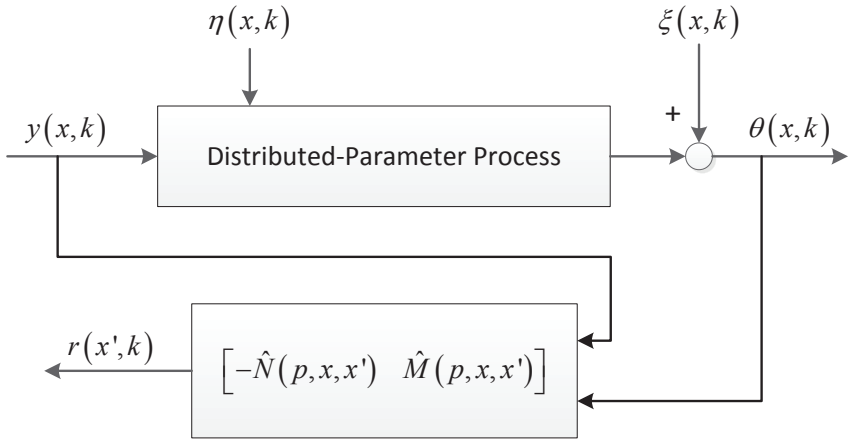


Figure 5.3: Kernel representation based residual generator for DPP

As shown in Figure 5.3, the kernel $\left[-\hat{\mathbf{N}}(p,x,x') \quad \hat{\mathbf{M}}(p,x,x')\right]$ represents redundancy of the DPP. But different from the scheme developed for LPP, the I/O "snapshot" data are manipulated by integration over space instead of multiplication.

Kernel representation for a common industrial configuration:

Nowadays, in production processes like the paper machine and the hot strip mill, the process variables are manipulated by lots of low-level components/subsystems while the state variables are measured at several fixed locations and are considered as technical KPIs. Under the assumption that KPIs are measured by ln_m sensors located at $\mathbf{x}_m = [x_{m,1}, \cdots, x_{m,n_m}]^T \in \mathcal{R}^{n_m}$, the lumped description of these processes can be simplified as

$$\begin{aligned}\mathbf{z}(k+1) &= \mathbf{A}_d\mathbf{z}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{E}_d(\hat{\mathbf{v}}(x), \boldsymbol{\eta}(x,k)),\\ \boldsymbol{\theta}(k) &= \mathbf{C}_{d,m}\mathbf{z}(k) + \boldsymbol{\xi}(k),\end{aligned}$$

where

$$\mathbf{C}_{d,m} = \begin{bmatrix} \hat{\mathbf{v}}_{x_m} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \hat{\mathbf{v}}_{x_m} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \hat{\mathbf{v}}_{x_m} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \in \mathcal{R}^{ln_m \times 2n\gamma},\ \hat{\mathbf{v}}_{x_m} = \begin{bmatrix} \hat{\mathbf{v}}(x_{m,1})^T \\ \vdots \\ \hat{\mathbf{v}}(x_{m,n_m})^T \end{bmatrix} \in \mathcal{R}^{n_m \times \gamma}$$

and $\boldsymbol{\xi}(k) \in \mathcal{R}^{ln_m}$ represents the measurement noise.

Consequently, the observer (5.36-5.37) can be reduced to

$$\hat{\mathbf{z}}(k+1) = (\mathbf{A}_d - \mathbf{L}\mathbf{C}_{d,m})\hat{\mathbf{z}}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{L}\boldsymbol{\theta}(k) \tag{5.47}$$

$$\hat{\boldsymbol{\theta}}(k) = \mathbf{C}_{d,m}\hat{\mathbf{z}}(k),\ \mathbf{r}(k) = \boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}(k) \tag{5.48}$$

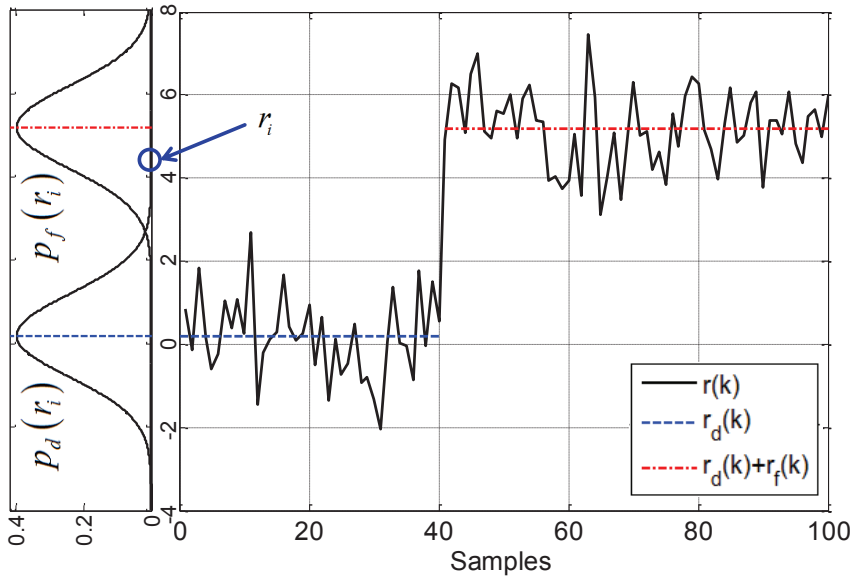


Figure 5.4: Statistical distribution of residual signal

where $\mathbf{L} \in \mathcal{R}^{2n\gamma \times ln_m}$ is a standard observer gain matrix. In this case, the kernel representation becomes

$$\mathbf{r} = \begin{bmatrix} -\hat{\mathbf{N}}(p) & \hat{\mathbf{M}}(p) \end{bmatrix} \begin{bmatrix} (\hat{\mathbf{v}}(x), \mathbf{y}(x)) \\ \boldsymbol{\theta} \end{bmatrix} \tag{5.49}$$

where

$$\hat{\mathbf{N}}(p) = \mathbf{C}_{d,m}(p\mathbf{I} - \mathbf{A}_d + \mathbf{L}\mathbf{C}_{d,m})^{-1}\mathbf{B}_d,$$
$$\hat{\mathbf{M}}(p) = \mathbf{I} - \mathbf{C}_{d,m}(p\mathbf{I} - \mathbf{A}_d + \mathbf{L}\mathbf{C}_{d,m})^{-1}\mathbf{L}.$$

5.3.3 Residual evaluation and threshold setting

The residual generated by either (5.36-5.37) or (5.47-5.48) provides a measure of discrepancy between the evolution of the actual DPP and the approximated finite dimensional description. Due to the projection-based lumping, the residual signal contains not only the impacts of random factors *i.e.* process and measurement noise, but also a deterministic part that is representing the projection error. In order to achieve satisfactory monitoring performance, a residual evaluation scheme will be established for the stochastic system with deterministic error. To this end, the dynamic equations of (5.36-5.37) and

(5.47-5.48) are reformulated as

$$\begin{aligned}\mathbf{e}(k+1) &= \bar{\mathbf{A}}_d\mathbf{e}(k) + \mathbf{E}_d\bar{\boldsymbol{\eta}}(k) - \bar{\boldsymbol{\xi}}(k) - \Delta\bar{\boldsymbol{\theta}}(k)\\ \mathbf{r}(x,k) &= \mathbf{C}_d(x)\mathbf{e}(k) + \boldsymbol{\xi}(x,k) + \boldsymbol{\Delta\theta}(x,k),\\ \bar{\boldsymbol{\xi}}(k) &= (\mathbf{L}(x), \boldsymbol{\xi}(x,k)), \boldsymbol{\Delta}\bar{\boldsymbol{\theta}} = (\mathbf{L}(x), \boldsymbol{\Delta\theta}(x,k)),\\ \mathbf{r}(k) &= ||\mathbf{r}(x,k)||\end{aligned}$$

and

$$\begin{aligned}\mathbf{e}(k+1) &= (\mathbf{A}_d - \mathbf{L}\mathbf{C}_{d,m})\mathbf{e}(k) + \mathbf{E}_d\bar{\boldsymbol{\eta}}(k) - \mathbf{L}\boldsymbol{\xi}(k) - \mathbf{L}\boldsymbol{\Delta\theta}(k)\\ \mathbf{r}(k) &= \mathbf{C}_{d,m}\mathbf{e}(k) + \boldsymbol{\xi}(k) + \boldsymbol{\Delta\theta}(k)\end{aligned}$$

where $\Delta\boldsymbol{\theta}(x,k)$ and $\Delta\boldsymbol{\theta}(k)$ represent the deterministic estimation error which is not contained in the reduced process description.

Assume that process and measurement noise is random distributed with zero mean values, then as shown in Figure 5.4, in the fault-free case, $\mathbf{r}_d(k)$ is non-zero. It is caused by projection error. In faulty case we have

$$E(\mathbf{r}(k)) = \mathbf{r}_d(k) + \mathbf{r}_f(k),\ \mathbf{r}(k) \in \mathcal{R}^{n_r}$$

where $\mathbf{r}_f(k)$ represents the impacts of faults. In the following, we will apply the GLR technique introduced in Chapter 2 to establish a residual evaluation scheme. For this purpose, based on given steady state residual $\mathbf{r}(k), i = 1, \cdots, N$, the GLR is computed as

$$\begin{aligned}2S_1^N &= 2\sum_{k=1}^{N}\log\frac{P_f(\mathbf{r}(k))}{P_d(\mathbf{r}(k))}\\ &= \sum_{k=1}^{N}(\mathbf{r}(k)-\mathbf{r}_d)^T\boldsymbol{\Sigma}^{-1}(\mathbf{r}(k)-\mathbf{r}_d) - \sum_{k=1}^{N}(\mathbf{r}(k)-\mathbf{r}_d-\mathbf{r}_f)^T\boldsymbol{\Sigma}^{-1}(\mathbf{r}(k)-\mathbf{r}_d-\mathbf{r}_f)\\ &= 2N\mathbf{r}_f^T\boldsymbol{\Sigma}^{-1}\bar{\mathbf{r}} - 2N\mathbf{r}_f^T\boldsymbol{\Sigma}^{-1}\mathbf{r}_d - N\mathbf{r}_f^T\boldsymbol{\Sigma}^{-1}\mathbf{r}_f,\ \bar{\mathbf{r}} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{r}(k),\\ &= N\bar{\mathbf{r}}^T\boldsymbol{\Sigma}^{-1}\bar{\mathbf{r}} - N\left((\bar{\mathbf{r}}-\mathbf{r}_f)^T\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{r}}-\mathbf{r}_f) + 2\mathbf{r}_f^T\boldsymbol{\Sigma}^{-1}\mathbf{r}_d\right)\end{aligned} \tag{5.50}$$

where

$$\begin{aligned}P_d(r_i) &= \frac{1}{\sqrt{(2\pi)^{n_r}\det(\boldsymbol{\Sigma})}}e^{\frac{1}{2}(\mathbf{r}(k)-\mathbf{r}_d)^T\boldsymbol{\Sigma}^{-1}(\mathbf{r}(k)-\mathbf{r}_d)},\\ P_f(r_i) &= \frac{1}{\sqrt{(2\pi)^{n_r}\det(\boldsymbol{\Sigma})}}e^{\frac{1}{2}(\mathbf{r}(k)-\mathbf{r}_d-\mathbf{r}_f)^T\boldsymbol{\Sigma}^{-1}(\mathbf{r}(k)-\mathbf{r}_d-\mathbf{r}_f)}.\end{aligned}$$

Algorithm 5.1. *Model-based KPI monitoring scheme for DPP*

S1: Determine the mesh size and the basis functions which span $\hat{\mathcal{V}}(x)$.
S2: Based on the original DPP model (5.1-5.2), derive the finite dimensional model (5.34-5.35).
S3: Design the residual generator (5.44-5.45) or (5.47-5.48) with properly selected observer gain matrix $\mathbf{L}(x)$ *or* $\mathbf{L}$.
S4: Based on the residual evaluation function (5.51), determine a threshold according to the noncentral χ^2*-distribution as (5.53).*
Based on the online data and the obtained parameters,
S5: Generate residual with (5.44-5.45) or (5.47-5.48), and evaluate it with (5.51).
S6: Check the decision logic (5.54).

From the above equation, we can see that the maximal likelihood estimate of fault is $\mathbf{r}_f = \bar{\mathbf{r}} - \mathbf{r}_d$ (by calculating the partial derivative of the second term of (5.50) with respect to $\mathbf{r}_f$ and setting it to zero). If a fault happens and a corresponding residual $\mathbf{r}(k)$ is generated, then as shown in Figure 5.4, the likelihood ratio $P_f(\mathbf{r}(k))/P_d(\mathbf{r}(k))$ will generally be greater than one. However, S_1^N is not suitable to be a test statistic since $\mathbf{r}_f$ is unknown *a prior*. For residual evaluation, we adopt the quadratic T^2 test statistic:

$$T^2 = N\left(\frac{1}{N}\sum_{k=1}^{N}\mathbf{r}(k)\right)^T \mathbf{\Sigma}^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\mathbf{r}(k)\right) \tag{5.51}$$

where $\mathbf{\Sigma}$ represents the covariance matrix of the *steady-state* residual vector and N is the length of evaluation window. If we have $S_1^N > 0$, or equivalently $N\bar{\mathbf{r}}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{r}} > (2\bar{\mathbf{r}} - \mathbf{r}_d)^T\mathbf{\Sigma}^{-1}\mathbf{r}_d$, then a decision saying that the process is faulty is likely to be made. In practice however, noise will make this decision making scheme fail to work due to too many false alarms.

In the following, we will use a robust decision marking scheme based on the statistical distribution of T^2. If the covariance matrix of noise is available, then $\mathbf{\Sigma}$ can be obtained by solving some static Lyapunov equations [44]. Otherwise, sufficient residual data $\mathbf{R} = \Big[\mathbf{r}(1), \cdots, \mathbf{r}(N_r)\Big]$ should be generated offline for estimating $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} \approx \frac{(\mathbf{R} - \mathbf{r}_d\mathbf{1}_{N_r}^T)(\mathbf{R} - \mathbf{r}_d\mathbf{1}_{N_r}^T)^T}{N_r - 1}, \mathbf{r}_d \approx \frac{1}{N_r}\sum_{k=1}^{N_r}\mathbf{r}(k). \tag{5.52}$$

Under normal operating condition, we have $\mathbf{r}(k) \sim \mathcal{N}(\mathbf{r}_d, \mathbf{\Sigma})$. As a result, the T^2 test statistic is noncentrally χ^2-distributed with n_r degrees of freedom and the noncentrality parameter is $N\mathbf{r}_d^T\mathbf{\Sigma}^{-1}\mathbf{r}_d$. The distribution of test statistic is

$$T^2 \sim \chi^2(n_r, N\mathbf{r}_d^T\mathbf{\Sigma}^{-1}\mathbf{r}_d)$$

For a given significance level α, the threshold is determined as

$$J_{th} = \chi^2_{1-\alpha}(n_r, \mathbf{r}_d^T \boldsymbol{\Sigma}^{-1} \mathbf{r}_d) \tag{5.53}$$

and the decision logic is as follows:

$$\text{If } T^2 > J_{th}, \text{ then faulty; otherwise, fault-free.} \tag{5.54}$$

The design steps based on given DPP model is summarized in Algorithm 5.1

5.4 Data-driven realization of KPI monitoring systems

In this section, we will realize the model-based KPI monitoring scheme in a data-driven manner. Available are following "snapshots" of the manipulated variables and the KPIs:

$$\mathbf{Y}(x) = \{\mathbf{y}(x,1), \cdots, \mathbf{y}(x,N)\}, \boldsymbol{\Theta}(x) = \{\boldsymbol{\theta}(x,1), \cdots, \boldsymbol{\theta}(x,N)\}.$$

Our first objective is to realize (5.46) in the time domain, which will serve as an initial residual generator.

Rewrite (5.36-5.37) as

$$\hat{\mathbf{z}}(k+1) = \mathbf{A}_d \hat{\mathbf{z}}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + (\mathbf{L}(x), \mathbf{r}(x,k)), \tag{5.55}$$

$$\boldsymbol{\theta}(x,k) = \mathbf{C}_d(x)\hat{\mathbf{z}}(k) + \mathbf{r}(x,k). \tag{5.56}$$

The dimension of $\boldsymbol{\theta}(x,k)$ can be reduced by projecting it onto the subspace $\hat{\mathcal{V}}(x)$, *i.e.*

$$\theta_i(x,k) \approx \text{proj}_{\hat{\mathcal{V}}(x)} \theta_i(x,k) = \hat{\mathbf{v}}^T(x)\bar{\bar{\boldsymbol{\theta}}}_i,$$
$$\bar{\bar{\boldsymbol{\theta}}}_i = (\hat{\mathbf{v}}(x), \hat{\mathbf{v}}^T(x))^{-1}(\hat{\mathbf{v}}(x), \theta_i(x,k)) \in \mathcal{R}^\gamma, \ i = 1, \cdots, l.$$

In the given subspace $\hat{\mathcal{V}}(x)$, the weighting vector $\bar{\bar{\boldsymbol{\theta}}}_i, i = 1, \cdots, l$, can be used to represent the KPI data $\boldsymbol{\theta}(x,k)$. Motivated by it, by (row-wise) multiplying both sides of (5.56) with $\hat{\mathbf{v}}(x) \in \mathcal{H}^\gamma$ and then integrating over space, we have

$$\begin{bmatrix} (\hat{\mathbf{v}}(x), \theta_1(x,k)) \\ \vdots \\ (\hat{\mathbf{v}}(x), \theta_l(x,k)) \end{bmatrix} = \begin{bmatrix} (\hat{\mathbf{v}}(x), r_1(x,k)) \\ \vdots \\ (\hat{\mathbf{v}}(x), r_l(x,k)) \end{bmatrix} + \begin{bmatrix} (\hat{\mathbf{v}}(x), \mathbf{c}_{d,1}(x)) \\ \vdots \\ (\hat{\mathbf{v}}(x), \mathbf{c}_{d,l}(x)) \end{bmatrix} \hat{\mathbf{z}}(k), \ \mathbf{C}_d(x) := \begin{bmatrix} \mathbf{c}_{d,1}(x) \\ \vdots \\ \mathbf{c}_{d,l}(x) \end{bmatrix}.$$

By introducing $\bar{\boldsymbol{\theta}}_i(k) = (\hat{\mathbf{v}}(x), \theta_i(x,k)), \bar{\mathbf{r}}_i(k) = (\hat{\mathbf{v}}(x), r_i(x,k))$ and $\bar{\mathbf{C}}_{d,i} = (\hat{\mathbf{v}}(x), \mathbf{c}_{d,i}(x))$, for $i = 1, \cdots, l$, (5.56) can be alternatively formulated as

$$\bar{\boldsymbol{\theta}}(k) = \bar{\mathbf{C}}_d \hat{\mathbf{z}}(k) + \bar{\mathbf{r}}(k)$$

where

$$\bar{\boldsymbol{\theta}}(k) = \begin{bmatrix} \bar{\boldsymbol{\theta}}_1(k) \\ \vdots \\ \bar{\boldsymbol{\theta}}_l(k) \end{bmatrix}, \ \bar{\mathbf{r}}(k) = \begin{bmatrix} \bar{\mathbf{r}}_1(k) \\ \vdots \\ \bar{\mathbf{r}}_l(k) \end{bmatrix}, \ \bar{\mathbf{C}}_d = \begin{bmatrix} \bar{\mathbf{C}}_{d,1} \\ \vdots \\ \bar{\mathbf{C}}_{d,l} \end{bmatrix} \in \mathcal{R}^{l\gamma \times 2n\gamma}.$$

In addition, according to (5.43), $(\mathbf{L}(x), \mathbf{r}(x,k))$ can be written as

$$(\mathbf{L}(x), \mathbf{r}(x,k)) = \begin{bmatrix} \mathbf{L}_{L,1} & \cdots & \mathbf{L}_{L,l} \end{bmatrix} \begin{bmatrix} (\hat{\mathbf{v}}(x), r_1(x,k)) \\ \vdots \\ (\hat{\mathbf{v}}(x), r_l(x,k)) \end{bmatrix}. \tag{5.57}$$

Based on it, an alternative residual generator in lumped form can be formulated as

$$\hat{\mathbf{z}}(k+1) = \mathbf{A}_d \hat{\mathbf{z}}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{L}_{\mathrm{L}} \bar{\mathbf{r}}(k) \tag{5.58}$$

$$\bar{\mathbf{r}}(k) = (\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k)) - \bar{\mathbf{C}}_d \hat{\mathbf{z}}(k), \tag{5.59}$$

where $(\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k)) = \bar{\boldsymbol{\theta}}(k)$.

It can be observed from the previous sections that the determination of the subspace $\hat{\mathcal{V}}(x)$ plays an essential role for establishing residual generators. Although the basis of the subspace can be easily selected by using the piece-wise polynomials with less process knowledge, the number of basis functions is generally unnecessarily high. In the following subsection, we will briefly introduce the basic idea of Karhumen-Loève (KL)-expansion, which is a popular dimension reduction method in the infinite dimensional domain and delivers a minimal number of basis functions for the subspace $\hat{\mathcal{V}}(x)$ based on the I/O "snapshot" data. The data-driven realization of (5.58-5.59) will be continued afterwards.

5.4.1 KL-expansion for optimal subspace selection

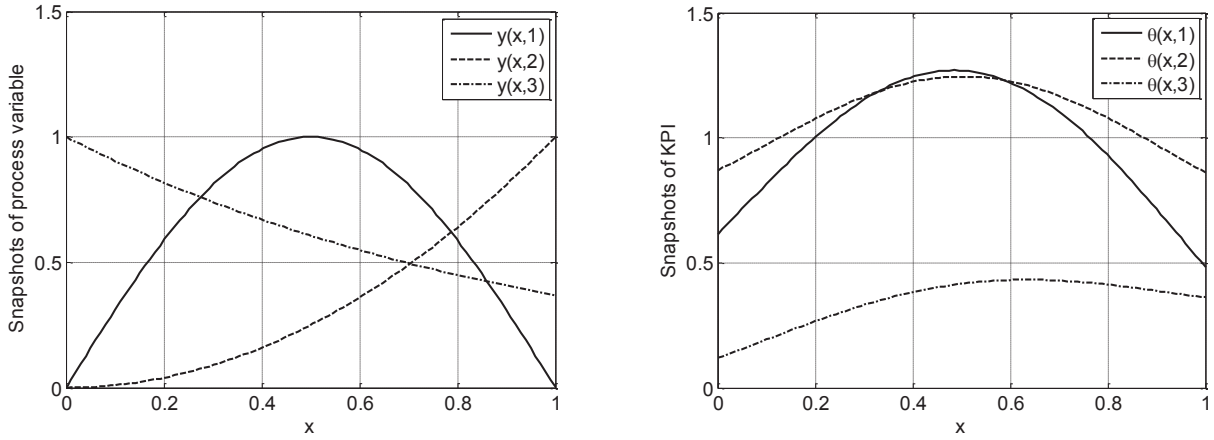
To determine a set of basis functions that span the available data, let us rearrange the I/O "snapshot" data as

$$\mathbf{d}(x,k) = \begin{bmatrix} y_1(x,k) & \cdots & y_m(x,k), & \theta_1(x,k) & \cdots & \theta_l(x,k) \end{bmatrix},$$

$$\boldsymbol{\Phi}(x) = \begin{bmatrix} \mathbf{d}(x,1) & \cdots & \mathbf{d}(x,N) \end{bmatrix} \in \mathcal{H}^{N_\phi \times 1}, N_\phi = (m+l)N$$

and denote

$$\boldsymbol{\Phi}(x) = \begin{bmatrix} \phi_1(x) & \cdots & \phi_{N_\phi}(x) \end{bmatrix} \in \mathcal{H}^{N_\phi \times 1}. \tag{5.60}$$



(a) Snapshots of process variable (b) Snapshots of KPI

Figure 5.5: Illustration of "snapshot" data

Suppose the basis function $v_i(x) \in \hat{\mathcal{V}}, i = 1, \cdots, \gamma$, can be represented by the "snapshot" data as

$$v_i(x) = \sum_{k=1}^{N_\phi} \psi_{k,i} \phi_k(x) \tag{5.61}$$

then the task is to determine the coefficient $\psi_{k,i}, k = 1, \cdots, N_\phi$. Following the description of [77], the basis function $v_i(x)$ that most closely match the "snapshot" data maximizes

$$\frac{1}{N_\phi} \sum_{k=1}^{N_\phi} ||(\phi_k(x), v_i(x)||^2 \tag{5.62}$$

with $||v_i(x)|| = 1$. The above optimization problem is not numerically direct solvable, thus the following function is defined

$$K(x, x') = \frac{1}{N_\phi} \sum_{k=1}^{N_\phi} \phi_k(x) \phi_k(x'). \tag{5.63}$$

Based on it, we have

$$
\begin{aligned}
\left(\left(K(x,x'),v_i(x')\right),v_i(x)\right) &= \left(\int_\alpha^\beta K(x,x')v_i(x')\mathrm{d}x', v_i(x)\right) \\
&= \int_\alpha^\beta \int_\alpha^\beta K(x,x')v_i(x')\mathrm{d}x' v_i(x)\mathrm{d}x \\
&= \frac{1}{N_\phi}\sum_{k=1}^{N_\phi}\int_\alpha^\beta \int_\alpha^\beta \phi_k(x)\phi_k(x')v_i(x')v_i(x)\mathrm{d}x'\mathrm{d}x \\
&= \frac{1}{N_\phi}\sum_{k=1}^{N_\phi}\left(\left(\phi_k(x),v_i(x)\right)\left(\phi_k(x'),v_i(x')\right)\right) \\
&= \frac{1}{N_\phi}\sum_{k=1}^{N_\phi}||(\phi_k(x),v_i(x))||^2.
\end{aligned}
$$

Then the problem of maximizing (5.62) is equivalent to the following eigenvalue problem

$$
\left(K(x,x'),v_i(x')\right) = \lambda_i v_i(x) \tag{5.64}
$$

subject to $||v_i(x)|| = 1$, where λ_i is the i-th eigenvalue. By submitting (5.61) and (5.63) into the above equation, we get

$$
\sum_{k=1}^{N_\phi}\phi_k(x)\left(\sum_{l=1}^{N_\phi}\left(\frac{1}{N_\phi}\int_\alpha^\beta \phi_k(x')\phi_l(x')\mathrm{d}x'\right)\psi_{l,i}\right) = \sum_{k=1}^{N_\phi}\phi_k(x)\lambda_i\psi_{k,i}. \tag{5.65}
$$

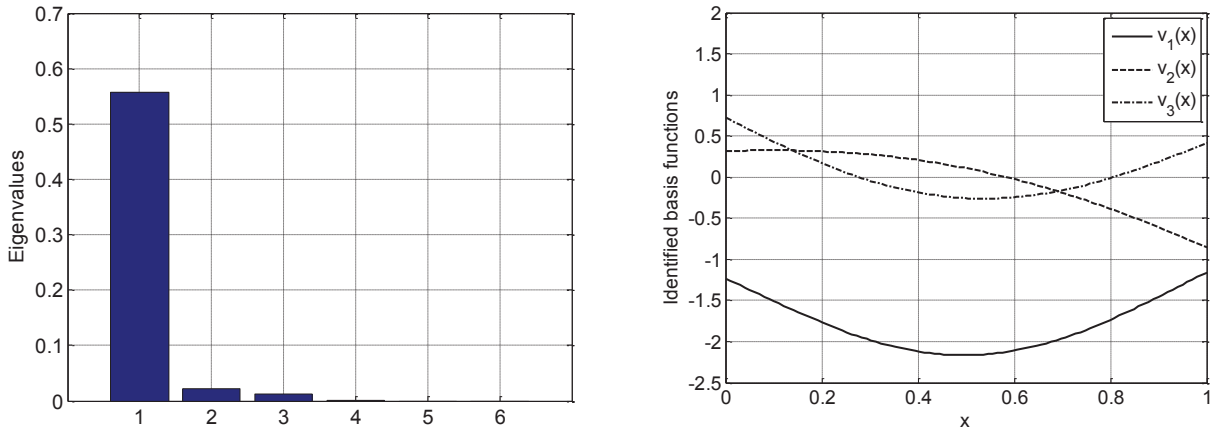
The largest eigenvalue λ_i can be found by solving the following matrix eigenvalue problem

$$
\begin{bmatrix}
\frac{1}{N_\phi}\int_\alpha^\beta \phi_1(x')\phi_1(x')\mathrm{d}x' & \cdots & \frac{1}{N_\phi}\int_\alpha^\beta \phi_1(x')\phi_{N_\phi}(x')\mathrm{d}x' \\
\vdots & \ddots & \vdots \\
\frac{1}{N_\phi}\int_\alpha^\beta \phi_{N_\phi}(x')\phi_1(x')\mathrm{d}x' & \cdots & \frac{1}{N_\phi}\int_\alpha^\beta \phi_{N_\phi}(x')\phi_{N_\phi}(x')\mathrm{d}x'
\end{bmatrix}
\begin{bmatrix}\psi_{1,i}\\ \vdots \\ \psi_{N_\phi,i}\end{bmatrix} = \lambda_i \begin{bmatrix}\psi_{1,i}\\ \vdots \\ \psi_{N_\phi,i}\end{bmatrix}. \tag{5.66}
$$

Denote $\bar{\boldsymbol{\psi}}_i = \begin{bmatrix}\psi_{1,i} & \cdots & \psi_{N_\phi,i}\end{bmatrix}^T$ as the eigenvector corresponding to the eigenvalue λ_i, then the optimal subspace can be determined by the first γ eigenvectors which contain the most information of the "snapshot" data (*e.g.* $\sum_{i=1}^{\gamma}\lambda_i/\sum_{i=1}^{N_\phi}\lambda_i > 1-\alpha$). The identified basis functions for the finite dimensional subspace is given as

$$
\hat{\mathbf{v}}(x) = \left[\sum_{j=1}^{N_\phi}\psi_{j,1}\phi_j(x), \cdots, \sum_{i=1}^{N_\phi}\psi_{j,\gamma}\phi_j(x)\right]^T = [v_1(x), \cdots, v_\gamma(x)]^T. \tag{5.67}
$$

Example 5.3. *In this example, as shown in Figure 5.5, the "snapshot" data for the process variable are given as* $\mathbf{y}(x) = \begin{bmatrix}sin(\pi x) & x^2 & e^{-x}\end{bmatrix}^T$ *and the KPI "snapshot" are*



(a) Ordered eigenvalues (b) Identified basis functions

Figure 5.6: KL expansion based subspace identification

given as $\boldsymbol{\theta}(x) = \text{rand}(3,3) \times \begin{bmatrix} sin(\pi x) & x^2 & e^{-x} \end{bmatrix}^T$. *The data are first rearranged using (5.60) and based on it, the data matrix from the left side of (5.66) is constructed. By doing an SVD of it, all eigenvalues and eigenvectors can be identified. As plotted in Figure 5.6-a, the first three eigenvalues are dominant and contain all the information. Thus we set* $\gamma = 3$ *and the corresponding basis functions are plotted in Figure 5.6-b. To briefly demonstrate its performance, we have projected both "snapshot" data onto the identified subspace. Figure 5.7 shows the projection error for all "snapshot" data. We can see that the identified subspace can well describe the "snapshot" data.*

Remark: Similar to the data-driven design scheme addressed in the previous chapter, the quality of "snapshot" data plays an important role for optimal subspace selection and further for performance of the realized monitoring system. In general, these "snapshot" data should be sufficiently excited and cover the whole normal operation region. Otherwise, the mesh size and basis functions should be manually determined according to basic process knowledge.

5.4.2 Data-driven realization of the kernel representation

In this subsection we will continue to discuss the lumped residual generator (5.58-5.59). In order to realize it in a data-driven way, we will use the following lumped data matrices

$$\bar{\mathbf{Y}}_{k,k+s} = \begin{bmatrix} (\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) & \cdots & (\hat{\mathbf{v}}(x), \mathbf{y}(x,k+N_c-1)) \\ \vdots & \ddots & \vdots \\ (\hat{\mathbf{v}}(x), \mathbf{y}(x,k+s)) & \cdots & (\hat{\mathbf{v}}(x), \mathbf{y}(x,k+s+N_c-1)) \end{bmatrix}, \tag{5.68}$$

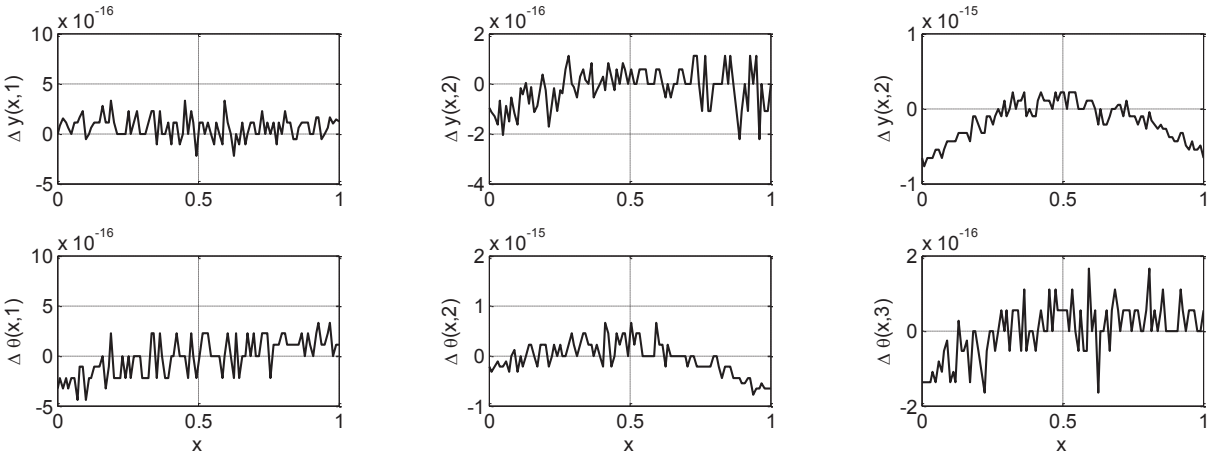


Figure 5.7: Projection error based on identified subspace

$$\bar{\mathbf{\Theta}}_{k,k+s} = \begin{bmatrix} (\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k)) & \cdots & (\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k+N_c-1)) \\ \vdots & \ddots & \vdots \\ (\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k+s)) & \cdots & (\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k+s+N_c-1)) \end{bmatrix}. \tag{5.69}$$

By following the procedures given in Chapter 4, the residual generator (5.58-5.59) can be extended to

$$\bar{\mathbf{\Theta}}_{k,k+s} = \begin{bmatrix} \bar{\mathbf{\Gamma}}_s \bar{\mathbf{\Phi}}_{s_p} & \bar{\mathbf{H}}_{y,s} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{\Omega}}_{k-s_p,k-1} \\ \bar{\mathbf{Y}}_{k,k+s} \end{bmatrix} + \bar{\mathbf{H}}_{r,s} \bar{\mathbf{R}}_{k,k+s} \tag{5.70}$$

where $\bar{\mathbf{\Gamma}}_s, \bar{\mathbf{\Phi}}_{s_p}, \bar{\mathbf{H}}_{y,s}, \bar{\mathbf{H}}_{r,s}$ have the same structure as in (4.18). Moreover, based on the following LQ-decomposition

$$\begin{bmatrix} \bar{\mathbf{\Omega}}_{k-s_p,k-1} \\ \bar{\mathbf{Y}}_{k,k+s} \\ \bar{\mathbf{\Theta}}_{k,k+s} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{L}}_{11} & \mathbf{0} & \mathbf{0} \\ \bar{\mathbf{L}}_{21} & \bar{\mathbf{L}}_{22} & \mathbf{0} \\ \bar{\mathbf{L}}_{31} & \bar{\mathbf{L}}_{23} & \bar{\mathbf{L}}_{33} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{Q}}_1^T \\ \bar{\mathbf{Q}}_2^T \\ \bar{\mathbf{Q}}_3^T \end{bmatrix}, \tag{5.71}$$

$$\bar{\mathbf{\Theta}}_{k,k+s} = \begin{bmatrix} \bar{\mathbf{L}}_{31} & \bar{\mathbf{L}}_{23} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{Q}}_1^T \\ \bar{\mathbf{Q}}_2^T \end{bmatrix} + \bar{\mathbf{L}}_{33} \bar{\mathbf{Q}}_3^T, \tag{5.72}$$

we have

$$\text{proj}_{\begin{bmatrix} \bar{\mathbf{\Omega}}_{k-s_p,k-1} \\ \bar{\mathbf{Y}}_{k,k+s} \end{bmatrix}}^{\perp} \bar{\mathbf{\Theta}}_{k,k+s} = \text{proj}_{\bar{\mathbf{Q}}_3^T} \bar{\mathbf{\Theta}}_{k,k+s}, \tag{5.73}$$

$$\Rightarrow \bar{\mathbf{H}}_{r,s} \bar{\mathbf{R}}_{k,k+s} (\bar{\mathbf{Q}}_3 \bar{\mathbf{Q}}_3^T) = \bar{\mathbf{L}}_{33} \bar{\mathbf{Q}}_3^T. \tag{5.74}$$

Since $(\bar{\mathbf{Q}}_3\bar{\mathbf{Q}}_3^T)^+ = (\bar{\mathbf{Q}}_3\bar{\mathbf{Q}}_3^T)$, we have the following approximation

$$\bar{\mathbf{H}}_{r,s}\bar{\mathbf{R}}_{k,k+s} \approx \bar{\mathbf{L}}_{33}\bar{\mathbf{Q}}_3^T(\bar{\mathbf{Q}}_3\bar{\mathbf{Q}}_3^T) = \bar{\mathbf{L}}_{33}\bar{\mathbf{Q}}_3^T. \tag{5.75}$$

Thus useful information about the lumped residual generator is mainly included in $\begin{bmatrix}\bar{\mathbf{L}}_{21} & \bar{\mathbf{L}}_{22}\\ \bar{\mathbf{L}}_{31} & \bar{\mathbf{L}}_{23}\end{bmatrix}$. By doing the following SVD,

$$\begin{bmatrix}\bar{\mathbf{L}}_{21} & \bar{\mathbf{L}}_{22}\\ \bar{\mathbf{L}}_{31} & \bar{\mathbf{L}}_{23}\end{bmatrix} = \begin{bmatrix}\bar{\mathbf{U}}_1 & \bar{\mathbf{U}}_2\end{bmatrix}\begin{bmatrix}\bar{\mathbf{\Lambda}}_1 & \mathbf{0}\\ \mathbf{0} & \bar{\mathbf{\Lambda}}_2 \approx \mathbf{0}\end{bmatrix}\begin{bmatrix}\bar{\mathbf{V}}_1^T & \bar{\mathbf{V}}_2^T\end{bmatrix}, \tag{5.76}$$

we have

$$\bar{\mathbf{\Psi}}_s^\perp = \begin{bmatrix}-\bar{\mathbf{\Psi}}_{s,y}^\perp & \bar{\mathbf{\Psi}}_{s,\theta}^\perp\end{bmatrix} = \bar{\mathbf{U}}_2^T. \tag{5.77}$$

As a result, the data-driven version of the kernel representation (5.6) is realized in the subspace $\hat{\mathcal{V}}(x)$ as

$$\mathbf{r}_s(k) = \begin{bmatrix}-\bar{\mathbf{\Psi}}_{s,y}^\perp & \bar{\mathbf{\Psi}}_{s,\theta}^\perp\end{bmatrix}\begin{bmatrix}(\hat{\mathbf{v}}(x),\mathbf{y}(x,k))\\ \vdots\\ (\hat{\mathbf{v}}(x),\mathbf{y}(x,k+s))\\ (\hat{\mathbf{v}}(x),\mathbf{\theta}(x,k))\\ \vdots\\ (\hat{\mathbf{v}}(x),\mathbf{\theta}(x,k+s))\end{bmatrix} \tag{5.78}$$

$$= \left(\begin{bmatrix}-\bar{\mathbf{\Psi}}_{s,y}^\perp(x) & \bar{\mathbf{\Psi}}_{s,\theta}^\perp(x)\end{bmatrix}, \begin{bmatrix}\mathbf{y}_s(x,k)\\ \mathbf{\theta}_s(x,k)\end{bmatrix}\right) \tag{5.79}$$

where

$$\begin{aligned}
&\mathbf{y}_s(x,k) = \begin{bmatrix}\mathbf{y}^T(x,k),\cdots,\mathbf{y}^T(x,k+s)\end{bmatrix}^T,\ \mathbf{\theta}_s(x,k) = \begin{bmatrix}\mathbf{\theta}^T(x,k),\cdots,\mathbf{\theta}^T(x,k+s)\end{bmatrix}^T,\\
&\bar{\mathbf{\Psi}}_{s,y}^\perp(x) = \begin{bmatrix}\bar{\mathbf{\Psi}}_{s,y,0}^\perp(x) & \cdots & \bar{\mathbf{\Psi}}_{s,y,s}^\perp(x)\end{bmatrix},\ \bar{\mathbf{\Psi}}_{s,\theta}^\perp(x) = \begin{bmatrix}\bar{\mathbf{\Psi}}_{s,\theta,0}^\perp(x) & \cdots & \bar{\mathbf{\Psi}}_{s,\theta,s}^\perp(x)\end{bmatrix},\\
&\bar{\mathbf{\Psi}}_{s,y,i}^\perp(x) = \begin{bmatrix}\bar{\mathbf{\Psi}}_{s,y,i}^\perp(:,1:\gamma)\hat{\mathbf{v}}(x) & \cdots & \bar{\mathbf{\Psi}}_{s,y,i}^\perp(:,(m-1)\gamma+1:m\gamma)\hat{\mathbf{v}}(x)\end{bmatrix},\\
&\bar{\mathbf{\Psi}}_{s,\theta,i}^\perp(x) = \begin{bmatrix}\bar{\mathbf{\Psi}}_{s,\theta,i}^\perp(:,1:\gamma)\hat{\mathbf{v}}(x) & \cdots & \bar{\mathbf{\Psi}}_{s,\theta,i}^\perp(:,(l-1)\gamma+1:l\gamma)\hat{\mathbf{v}}(x)\end{bmatrix},\ i=0,\cdots,s
\end{aligned} \tag{5.80}$$

with

$$\bar{\mathbf{\Psi}}_{s,y,i}^\perp = \bar{\mathbf{\Psi}}_{s,y}^\perp(:,im\gamma+1:(i+1)m\gamma), \bar{\mathbf{\Psi}}_{s,\theta,i}^\perp = \bar{\mathbf{\Psi}}_{s,\theta}^\perp(:,il\gamma+1:(i+1)l\gamma),\ i=0,\cdots,s.$$

The above kernel representation serves as a primary residual generator for the original DPP. For residual evaluation, the following T^2 test statistic is used:

$$T^2 = \mathbf{r}_s^T\mathbf{\Sigma}_{r,s}^{-1}\mathbf{r}_s \tag{5.81}$$

Algorithm 5.2. *Data-driven realization of kernel representation of DPPs*

Based on the offline "snapshot" data $\mathbf{Y}(x)$ *and* $\mathbf{\Theta}(x)$*,*

S1: Determine the basis functions $v_1(x), \cdots, v_\gamma(x)$ *using KL-expansion or manually.*

S2: Constructed the lumped data matrices according to (5.68-5.69).

S3: Identify $\bar{\mathbf{\Psi}}_s^\perp$ *according to (5.77).*

S4: Build $\bar{\mathbf{\Psi}}_{s,y}^\perp(x)$ *and* $\bar{\mathbf{\Psi}}_{s,\theta}^\perp(x)$ *using (5.80) and determine the threshold using (5.82).*

Based on the online "snapshot" data $\mathbf{y}_s(x,k)$ *and* $\mathbf{\theta}_s(x,k)$*,*

S5: Construct the residual generator (5.79) and build the test statistic (5.81).

S6: Check the decision logic (5.54).

where $\mathbf{\Sigma}_{r,s} \approx \bar{\mathbf{\Psi}}_{s,\theta}^\perp \bar{\mathbf{L}}_{33} \bar{\mathbf{L}}_{33}^T (\bar{\mathbf{\Psi}}_{s,\theta}^\perp)^T/(N_c-1)$ or can be alternatively estimated using fault-free residual data. In case that $\mathbf{\Sigma}_{r,s}$ is rank deficient, the same technique given in (4.31) should be used. The threshold can be determined using the noncentral χ^2-distribution as

$$J_{th} = \chi^2_{1-\alpha}(n_s, E(\mathbf{r})^T \mathbf{\Sigma}_{r,s}^{-1} E(\mathbf{r})) \tag{5.82}$$

where $n_s = size(\bar{\Psi}_s^\perp, 1)$.

The procedures for data-driven design of the kernel representation (5.6) are given in Algorithm 5.2.

For the common industrial configuration:

As discussed in the previous section, distributed manipulation and lumped KPI measuring is a common industrial configuration. A model-based residual generator has been designed for it in (5.47-5.48), *i.e.*

$$\begin{aligned} \hat{\mathbf{z}}(k+1) &= (\mathbf{A}_d - \mathbf{L}\mathbf{C}_{d,m})\hat{\mathbf{z}}(k) + \mathbf{B}_d(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{L}\mathbf{r}(k), \\ \mathbf{r}(k) &= \mathbf{\theta}(k) - \mathbf{C}_{d,m}\hat{\mathbf{z}}(k). \end{aligned}$$

To realize it in the data-driven framework, we need to construct the same lumped process data as (5.68) and build the KPI data $\bar{\mathbf{\Theta}}_{k,k+s}$ as

$$\bar{\mathbf{\Theta}}_{k,k+s} = \begin{bmatrix} \mathbf{\theta}(k) & \cdots & \mathbf{\theta}(k+N_c-1) \\ \vdots & \vdots & \vdots \\ \mathbf{\theta}(k+s) & \cdots & \mathbf{\theta}(k+s+N_c-1) \end{bmatrix} \in \mathcal{R}^{(s+1)l \times N_c}, \tag{5.83}$$

where the original KPI data is denoted as $\mathbf{\Theta} = [\mathbf{\theta}_k, \cdots, \mathbf{\theta}(k+N-1)], N >> s+N_c$.

Based on them, we could identify $\bar{\mathbf{\Psi}}_s^\perp$ according to $S3$ from Algorithm 5.2. Finally the following residual generator is achieved:

$$\mathbf{r}_s(k) = \bar{\mathbf{\Psi}}_{s,\theta}^\perp \mathbf{\theta}_s(k) - \left(\bar{\mathbf{\Psi}}_{s,y}^\perp(x), \mathbf{y}_s(x,k)\right) \tag{5.84}$$

Algorithm 5.3. *Data-driven realization of kernel representation of DPPs for a common industrial configuration*

Based on the offline "snapshot" data $\mathbf{Y}(x)$ *and the KPI data* $\mathbf{\Theta}$*,*

S1: Determine the basis functions $v_1(x), \cdots, v_\gamma(x)$ *using KL-expansion or manually.*

S2: Constructed the lumped process data matrix according to (5.68) and the KPI data matrix as (5.83).

S3: Identify $\bar{\mathbf{\Psi}}_s^\perp$ *according to (5.77).*

S4: Build $\bar{\mathbf{\Psi}}_{s,y}^\perp(x)$ *according to (5.80) and determine the threshold using (5.82).*

Based on the online process "snapshot" data $\mathbf{y}_s(x,k)$ *and the KPI measurement* $\mathbf{\theta}_s(k)$*,*

S5: Construct the residual generator (5.84) and build the test statistic (5.81).

S6: Check the decision logic (5.54).

where $\bar{\mathbf{\Psi}}_{s,y}^\perp(x)$ is built according to (5.80) and $\mathbf{\theta}_s(k) = [\mathbf{\theta}^T(k), \cdots, \mathbf{\theta}^T(k+s)]^T$.

The whole design procedures for this common configuration is summarized in Algorithm 5.3.

5.4.3 Observer-based implementation scheme

The design procedure of observer-based implementation scheme follows the same procedures as in Chapter 4. Let $[\boldsymbol{\alpha}_s, \boldsymbol{\beta}_s]$ be any vector belonging to the row space of $\bar{\mathbf{\Psi}}_s^\perp$ and denote

$$\boldsymbol{\alpha}_s = [\boldsymbol{\alpha}_{s,0}, \boldsymbol{\alpha}_{s,1}, \cdots, \boldsymbol{\alpha}_{s,s}],\ \boldsymbol{\beta}_s = [\boldsymbol{\beta}_{s,0}, \boldsymbol{\beta}_{s,1}, \cdots, \boldsymbol{\beta}_{s,s}] \tag{5.85}$$

where $\boldsymbol{\alpha}_{s,i} \in \mathcal{R}^{1\times(s+1)m\gamma}, \boldsymbol{\beta}_{s,i} \in \mathcal{R}^{1\times(s+1)l\gamma}, i = 0, \cdots, s$.

By running $S2$ of algorithm 4.2, we can obtain the following observer

$$\hat{\mathbf{z}}(k+1) = \mathbf{G}\hat{\mathbf{z}}(k) + \mathbf{H}(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{L}(\hat{\mathbf{v}}(x), \mathbf{\theta}(x,k)) \tag{5.86}$$

$$r(k) = \mathbf{v}(\hat{\mathbf{v}}(x), \mathbf{\theta}(x,k)) - \mathbf{w}\hat{\mathbf{z}}(k) - \mathbf{q}(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) \tag{5.87}$$

For residual evaluation, the T^2 statistic is established as

$$T^2 = r^2/\sigma_r^2 \tag{5.88}$$

where $\sigma_r^2 = \frac{1}{N-1}\sum_{k=1}^N (r(k) - \frac{1}{N}\sum_{k=1}^N r(k))^2$.

The threshold is determined using noncentral χ^2-distribution as

$$J_{th} = \chi_{1-\alpha}^2(1, E(r)^2/\sigma_r^2). \tag{5.89}$$

In Algorithm 5.4, the design of observer-based implementation scheme is summarized.

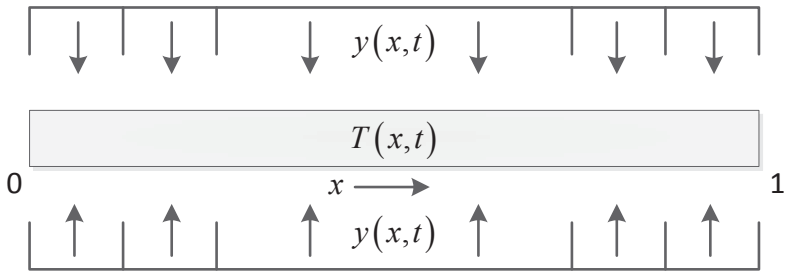


Figure 5.8: A long thin rod being heated in a multizone furnace [95]

Algorithm 5.4. *Data-driven realization of diagnostic observer for DPPs*

S1: Run algorithm (5.2) to get $\bar{\Psi}_s^{\perp}$.
S2: Construct $\boldsymbol{\alpha}_s, \boldsymbol{\beta}_s$ *and run S2 of Algorithm 4.2 to get the parameters of (5.86-5.87).*
S3: Determine the threshold using (5.89).
Based on the online "snapshot" data $\mathbf{y}(x,k)$ *and* $\boldsymbol{\theta}(x,k)$,
S4: Compute $(\hat{\mathbf{v}}(x), \mathbf{y}(x,k))$ *and* $(\hat{\mathbf{v}}(x), \boldsymbol{\theta}(x,k))$
S5: Generate the residual as (5.86-5.87) and build the T^2 *statistic as (5.88).*
S6: Check the decision logic (5.54).

For the common industrial configuration:

For the common industrial configuration the observer (5.86-5.87) is simplified as

$$\hat{\mathbf{z}}(k+1) = \mathbf{G}\hat{\mathbf{z}}(k) + \mathbf{H}(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) + \mathbf{L}\boldsymbol{\theta}(k) \tag{5.90}$$

$$r(k) = \mathbf{v}\boldsymbol{\theta}(k) - \mathbf{w}\hat{\mathbf{z}}(k) - \mathbf{q}(\hat{\mathbf{v}}(x), \mathbf{y}(x,k)) \tag{5.91}$$

The design procedures are the same as Algorithm 5.4 except for $S4 - S5$, which are:

Based on the online "snapshot" data $\mathbf{y}(x,k)$ *and the KPI data* $\boldsymbol{\theta}(k)$,
S4: Compute $(\hat{\mathbf{v}}(x), \mathbf{y}(x,k))$,
S5: Generate residual as (5.90-5.91) $\cdots$

5.5 Numerical examples

In this section, we consider the example given in [95, p. 149]. As shown in Figure 5.8, a long, thin rod is heated in a multizone furnace. Its temperature distribution $z(x,t)$ is controlled by manipulating the heating rate $y(x,t)$. It is assumed that both the manipulated variable and the temperature (considered as the KPI) can be measurable along the

whole x. The (dimensionless) mathematical model of this process is

$$\frac{\partial z(x,t)}{\partial t} = \frac{\partial^2 z(x,t)}{\partial x^2} + y(x,t) + \eta(x,t),$$
$$\theta(x,t) = z(x,t) + \xi(x,t),\ 0 \leq x \leq 1, t \geq 0$$

subject to

$$\left.\frac{\partial z(x,t)}{\partial x}\right|_{x=0,1} = 0.$$

In this example, the manipulated variable is set to be

$$y(x,k) = \begin{cases} 3sin(\pi x)(1-e^{-k}) + \epsilon(x,k), & k = 1, \cdots, 200; \\ 3sin(\pi x)e^{-k+200} + \epsilon(x,k), & k = 201, \cdots, 2000 \end{cases}$$

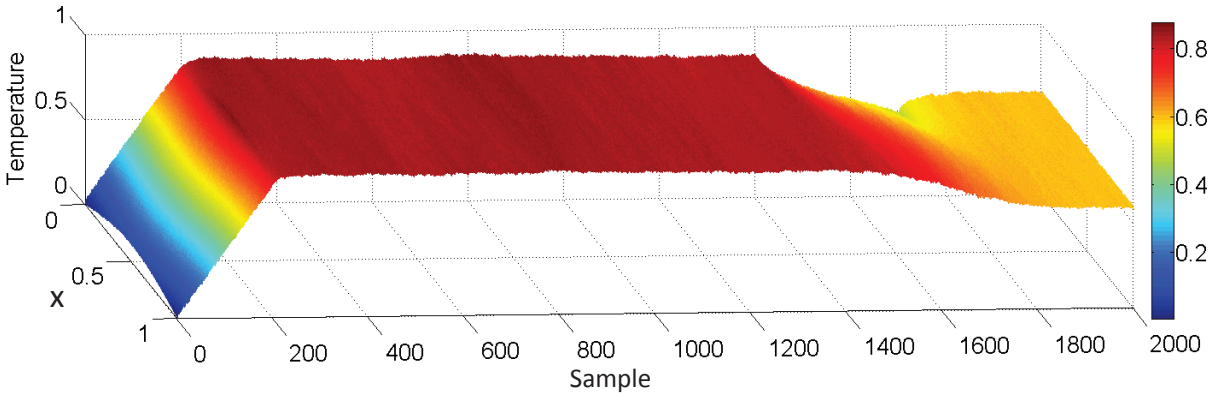
where $\epsilon(x,k) \sim \mathcal{N}(0, 1e-4)$. For simulating the DPP, the projection-based model reduction technique is used where 101 polynomial basis functions are defined. Figure 5.9-a to Figure 5.9-c show the temperature evolution of the rod. The initial temperature distribution can be seen from Figure 5.9-a and Figure 5.9-b at sample number 1, where the middle of the rod is warmer than both sides. With heat flowing into the rod, the temperature continuously increases. Constrained by the Neumann boundary conditions, both sides of the rod are perfectly insulated and no heat flows outside them. As can be observed from both figures, at about the 300th sample, the process arrives at the steady state. To demonstrate the proposed monitoring algorithms, a fault is introduced during the 1401st-1700th samples. This fault simulates the malfunctions that make heat flow out of the rod from the left side. As given in Figure 5.9-a and Figure 5.9-c, the temperature decreases significantly during that period. Since no further heat flows into the rod afterwards, the temperature arrives at a new steady state which is lower than the previous one.

In order to test the proposed methods for the common industrial configuration, we assume four temperature sensors are located at $x_m = [0.1, 0.4, 0.6, 0.9]^T$. Then the measurement equation becomes

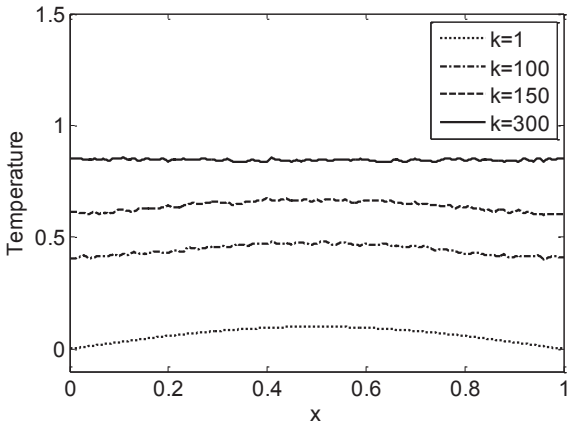
$$\theta(t) = \int_0^1 \Delta(x - x_m)(x,t)\mathrm{d}x + \bar{\boldsymbol{\xi}}(k)$$

where

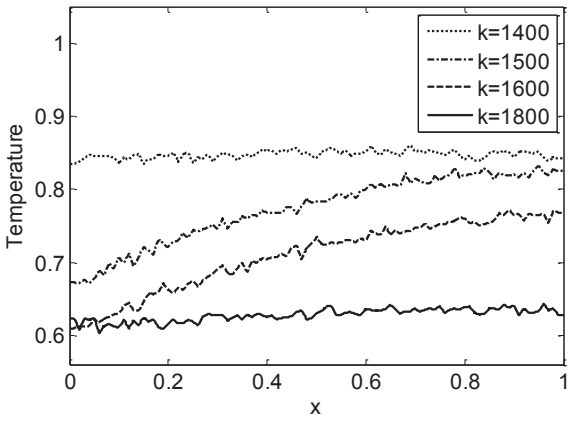
$$\Delta(x - x_m) = \begin{bmatrix} \delta(x-0.1) \\ \delta(x-0.4) \\ \delta(x-0.6) \\ \delta(x-0.9) \end{bmatrix}.$$



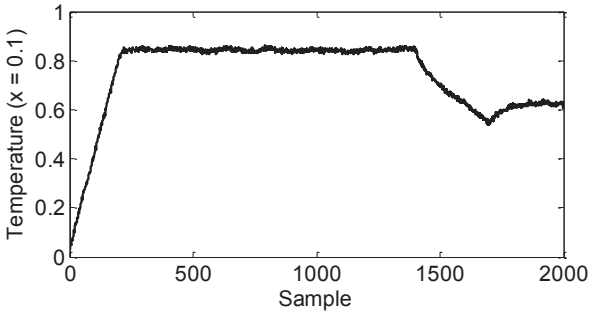
(a) Temperature evolution of the thin rod



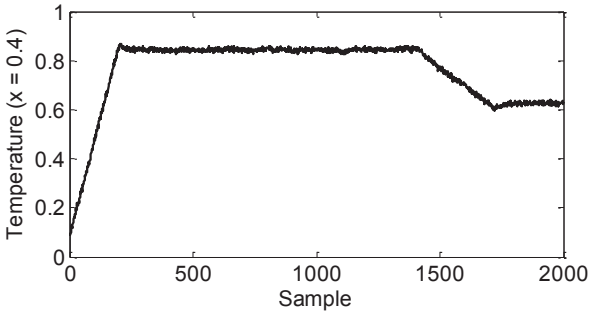
(b) Snapshots of KPI during the start-up phase



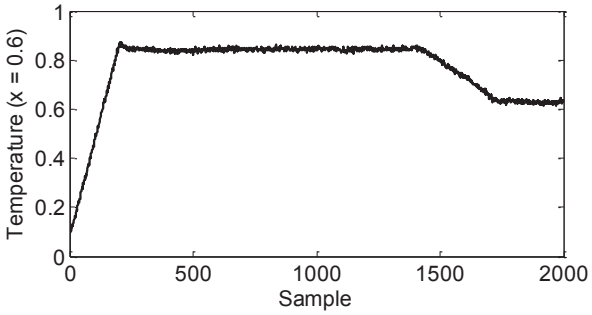
(c) Snapshots of KPI after fault happens



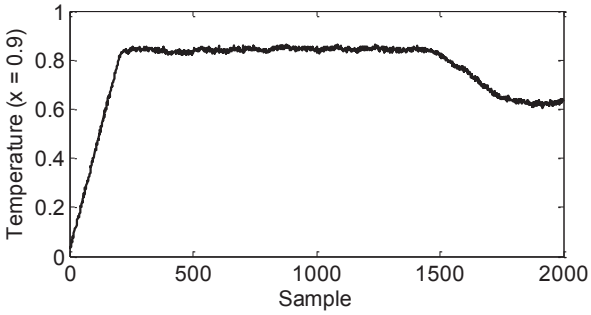
(d) KPI measurement at x = 0.1



(e) KPI measurement at x = 0.4



(f) KPI measurement at x = 0.6



(g) KPI measurement at x = 0.9

Figure 5.9: Temperature evolution of the long thin rod in a simulation

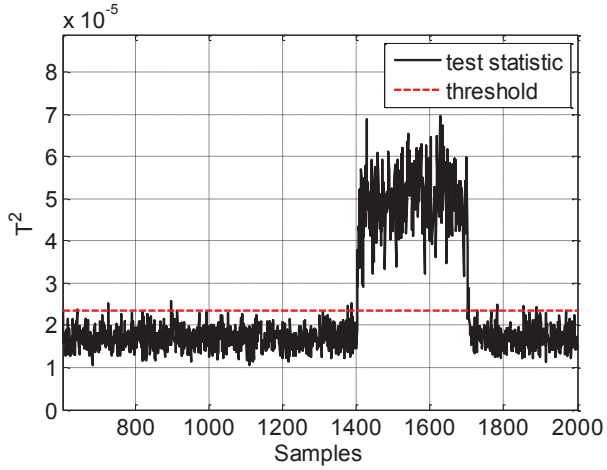


Figure 5.10: Model-based KPI monitoring result with measurable KPI snapshot

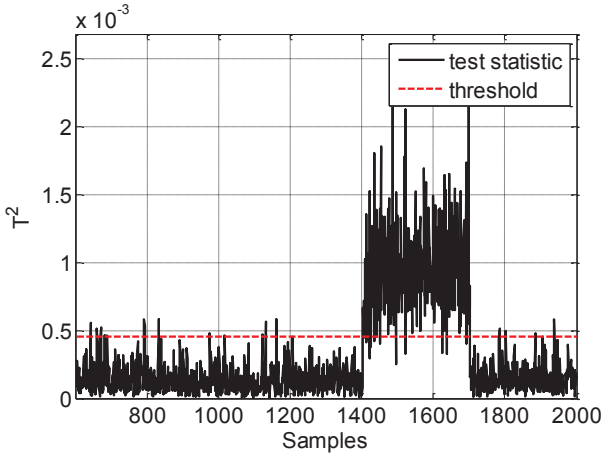


Figure 5.11: Model-based KPI monitoring result with 4 KPI sensors

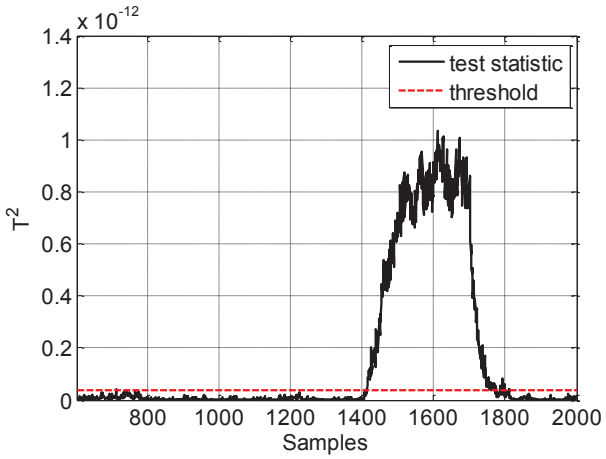


Figure 5.12: Data-driven KPI monitoring result with measurable KPI snapshot

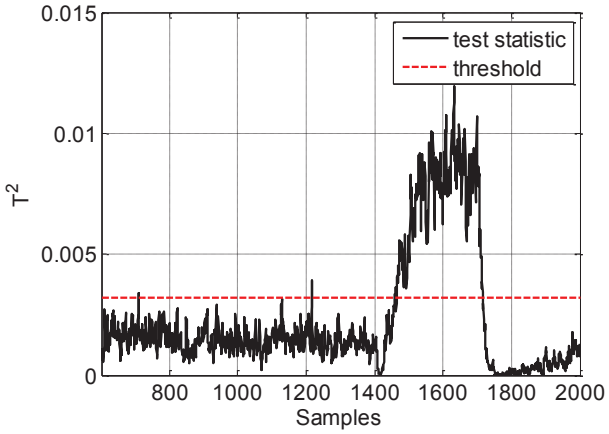


Figure 5.13: Data-driven KPI monitoring result with 4 KPI sensors

Figure 5.9-d to Figure 5.9-g show the KPI measurements given by these four sensors. We can see that the closer the sensor is to the left side, the more sensitive they are to the fault.

We first test Algorithm 5.1 with measurable KPI snapshot. Eleven polynomial basis functions (detailed description can be found in our early publication [45]) are used to construct the residual generator. By integrating the Neumann boundary conditions, the order of residual generator becomes 9. We have designed $\mathbf{l}(x)$ in such a way that eigenvalues of $(\mathbf{A}_d - (\mathbf{l}(x), \mathbf{c}_d(x)))$ are all 0.01. Figure 5.10 shows the monitoring result from the 601st sample. The 201st-600th samples are used to construct the test statistic and to determine the threshold during the offline phase. We can see that the heat leakage has been successfully monitored. Similarly, instead of using KPI snapshot, Figure 5.11 gives KPI monitoring result based on the four KPI sensors described above. The fault magnitude is enlarged by 5 times and the heat leakage has been detected as well. By comparing these results, we can see that the KPI monitoring system based on the snapshot data is more sensitive. The reason is that the snapshot data contain much more information about the fault than the four sensors.

Based on I/O snapshot data, we have determined five basis functions based on the KL-expansion method, which captures more than 99% variability of the data. The online I/O snapshot data are then lumped on these identified basis functions. For identifying the kernel representation, we choose $s = 3$ and $s_p = 5$. Figure 5.12 gives the monitoring result. We can see that the fault has been detected. For demonstrating the data-driven KPI monitoring algorithm for the common industrial configuration, we choose $s = 4$ and $s_p = 5$. The monitoring result is given in Figure 5.13. It can be observed that the fault can be successfully detected as well. Nevertheless, the sensitivity based on the KPI snapshot data is higher.

5.6 Concluding remarks

In this chapter, some novel solutions for performance monitoring in DPPs have been proposed. Based on the concept of projection in infinite dimensional space, a distributed kernel representation of the original DPP is achieved. It represents the analytical redundancy and serves as an initial residual generator. Then, a GLR-based residual revaluation scheme is established for alarm generation. Considering the common industrial configuration that the process variables are manipulated by lots of low-level components while the KPIs are measured at several given locations, an alternative monitoring scheme has been proposed. Since KPI "snapshot" data contain much richer information than the distributed KPI sensor data, the monitoring performance is better as well. However, this

scheme requires the online KPI "snapshots". Aiming at reducing the design effort for industrial application, the aforementioned schemes have been realized in a data-driven way. Based on available I/O "snapshots" data, a set of basis functions with lower dimension is identified. Then by using the basic idea of the identification technique discussed in Chapter 4, data-driven design of residual generators from the lumped data has been realized. All developed schemes have been summarized in the form of algorithms and their effectiveness is illustrated through numerical examples. In the next chapter, we will study data-driven diagnosis issue of performance degradation. For those readers who are interested in process monitoring with varying system parameters and strong nonlinearities, the early work [42, 84, 121] from our group are recommended.

6 Data-driven diagnosis of multiplicative fault

After key performance degradation is detected, it is desirable to diagnose and identify the source(s) of the fault and apply necessary corrective actions. Diagnosis of multiplicative fault plays an essential role in automation processes, especially for OEE improvement. The well-known fault diagnosis methods include fisher discriminant analysis [17], structural residual based approach [38] and contribution plots [81, 114]. The first two approaches demand sufficient *a priori* knowledge about the fault, which is generally difficult to obtain in large-scale processes. The third approach is very efficient and has been widely applied in practice, however, misdiagnosis might be caused by the so-called fault "smearing" effect [91, 114]. Recently, the reconstruction-based contribution (RBC) plots [2] scheme is proposed as a revised approach for the traditional contribution plots. This scheme diagnoses the fault by reconstructing the test statistic along each fault direction. It assists process engineers by identifying the variables that are closely related to the faulty component thus greatly narrows down the investigation scope.

Fault diagnosis based on RBC plots relies on an additive fault model. As discussed in Chapter 2, this type of fault does not influence the variance/covariance of process data. In practice, however, many technical components/loops in large-scale processes may frequently be subject to multiplicative faults. One example is the increased variability of process variables that are probably caused by wearing and aging of components or decreasing (loop) control performance. These faults could either cause direct economical losses by producing low quality products or shorten components' service life through very active movements. Motivated by these observations, a new data-driven multiplicative fault diagnosis scheme is developed [46]. Using the offline trained parameters and the online data, the fault features are firstly extracted. Then, the risky component(s) can be identified by evaluating the impacts of the fault on the test statistic along the corresponding variable direction or subspace. The proposed scheme is suitable for complex processes where manual investigations are either too costly or time-consuming. The objective is to increase the OEE by narrowing down the investigation scope.

6.1 Preliminaries and problem formulation

In this section, we will first introduce the basic idea of an alternative scheme to the RBC-based additive fault diagnosis method. The notations are consistent with Chapter 3. Similar to the RBC-based method, the alternative approach is based on the following additive fault model

$$\mathbf{y}_{af} = \mathbf{y} + \boldsymbol{\xi}_i f \tag{6.1}$$

where $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_y)$ denotes the normalized fault-free process vector, $\boldsymbol{\xi}_i = \mathbf{I}_m(:, i)$, for $i = 1, \cdots, m$, denotes the direction of fault, and $f \in \mathcal{R}$ denotes the fault magnitude. It is clear that the mean vector of faulty variables is influenced, *i.e.* $\mathbf{y}_{af} \sim \mathcal{N}(\boldsymbol{\xi}_i f, \boldsymbol{\Sigma}_y)$. Once performance degradation is detected, we want to identify the contribution of each variable (group) to it. By ranking them, the risky component(s) can be identified.

Available are the normalized process data $y(k),$ for $k = 1, \cdots, N$. For fault estimation, we establish the following GLR test

$$\begin{aligned} S_1^N =& \sum_{k=1}^{N} \ln \frac{P_{af}(\mathbf{y}(k))}{P(\mathbf{y}(k))} = \frac{1}{2} \sum_{k=1}^{N} \left(\mathbf{y}^T(k) \boldsymbol{\Sigma}_y^{-1} \mathbf{y}(k) - (\mathbf{y}(k) - \boldsymbol{\xi}_i f)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y}(k) - \boldsymbol{\xi}_i f) \right) \\ =& \sum_{k=1}^{N} \mathbf{y}^T(k) \boldsymbol{\Sigma}_y^{-1} \boldsymbol{\xi}_i f - \frac{N}{2} (\boldsymbol{\xi}_i f)^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{\xi}_i f \end{aligned} \tag{6.2}$$

where

$$\begin{aligned} P(\mathbf{y}(k)) &= \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}_y|}} \mathrm{e}^{\left(-\frac{1}{2} \mathbf{y}^T(k) \boldsymbol{\Sigma}_y^{-1} \mathbf{y}(k)\right)}, \\ P_{af}(\mathbf{y}(k)) &= \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}_\mathbf{y}|}} \mathrm{e}^{\left(-\frac{1}{2} (\mathbf{y}(k) - \boldsymbol{\xi}_i f)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y}(k) - \boldsymbol{\xi}_i f)\right)}. \end{aligned}$$

The $\boldsymbol{\xi}_i f$ which maximizes the GLR defined in (6.2) is the $\boldsymbol{\xi}_i f$ which solves the following linear equation system

$$\frac{\partial S_1^N}{\partial (\boldsymbol{\xi}_i f)} = \sum_{k=1}^{N} \mathbf{y}^T(k) \boldsymbol{\Sigma}_y^{-1} - N (\boldsymbol{\xi}_i f)^T \boldsymbol{\Sigma}_y^{-1} = \mathbf{0}. \tag{6.3}$$

As a result, the maximum likelihood estimate (MLE) of $\boldsymbol{\xi}_i f$ is given as

$$\boldsymbol{\xi}_i \hat{f} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{y}(k) \Rightarrow \hat{f} \approx \boldsymbol{\xi}_i^T \frac{1}{N} \sum_{k=1}^{N} \mathbf{y}(k). \tag{6.4}$$

For simplicity, here we only consider the T^2 test statistic (3.9) for KPI monitoring. Define $\mathbf{\Omega}_{\hat{\theta}} = \mathbf{P}_1\mathbf{\Lambda}_1^{-1/2}\mathbf{S}_1\mathbf{S}_1^T\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T$, by substituting (6.4) into (3.9), we have

$$T^2_{\hat{\theta},af} = (\mathbf{y} - \boldsymbol{\xi}_i\hat{f})^T\mathbf{\Omega}_{\hat{\theta}}(\mathbf{y} - \boldsymbol{\xi}_i\hat{f}) + 2\mathbf{y}^T\mathbf{\Omega}_{\hat{\theta}}\boldsymbol{\xi}_i\hat{f} - (\boldsymbol{\xi}_i\hat{f})^T\mathbf{\Omega}_{\hat{\theta}}\boldsymbol{\xi}_i\hat{f}$$

where the first term on the right side represents the reconstructed fault-free test statistic under the assumption that the fault happens along $\boldsymbol{\xi}_i$,

$$T^2_R := (\mathbf{y} - \boldsymbol{\xi}_i\hat{f})^T\mathbf{\Omega}_{\hat{\theta}}(\mathbf{y} - \boldsymbol{\xi}_i\hat{f})$$

while the remaining two represent fault contribution and are denoted as

$$\Delta T^2_{\hat{\theta},af,i} = 2\mathbf{y}^T\mathbf{\Omega}_{\hat{\theta}}\boldsymbol{\xi}_i\hat{f} - (\boldsymbol{\xi}_i\hat{f})^T\mathbf{\Omega}_{\hat{\theta}}\boldsymbol{\xi}_i\hat{f}.$$

As a result, in the faulty case, the test statistic contains information about the normal process variation and the fault. Based on the estimation of $\hat{f}$, the contribution of the fault to the test statistic can be constructed using the above equation. And it satisfies

$$\Delta T^2_{\hat{\theta},af,i} = T^2_{\hat{\theta},af} - T^2_R.$$

By plotting the average of $\Delta T^2_{\hat{\theta},af,i}$, for $i = 1, \cdots, m$, *i.e.*

$$E(\Delta T^2_{\hat{\theta},af,i}) = \bar{\mathbf{y}}^T(2\mathbf{I} - \boldsymbol{\xi}_i\boldsymbol{\xi}_i^T)\mathbf{\Omega}_{\hat{\theta}}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T\bar{\mathbf{y}},\ \bar{\mathbf{y}} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{y}(k),$$

the variable(s) that is(are) related to the (additive) KPI degradation can be identified.

As discussed previously, multiplicative faults frequently happen in the industrial practice. Different from the additive faults which change the mean vectors, they generally influence covariance matrices and thus are much harder to diagnose. The focus of this chapter is to study the diagnosis issue of the multiplicative fault

$$\mathbf{y}_{mf} = \mathbf{F}\mathbf{y},\ \mathbf{F} \neq \mathbf{I}_m \tag{6.5}$$

which increases the variability of the low-level process variables. According to their impacts, multiplicative faults occurred in the process can be classified into two groups, *i.e.* KPI-related faults and KPI-unrelated faults. Those KPI-related faults could cause direct economic losses due to decreased product quality, production efficiency, *etc.* Although the KPI-unrelated faults may not cause direct economic losses, they should be taken into consideration as well since service life of components might be shortened thus cause losses indirectly. From the monitoring and especially diagnosis point of view, the KPI-related faults are further divided into

- KPI-related faults in the low-level "process variable space" and
- KPI-related faults in the normalized "coefficient space".

Based on the reformulated process model

$$\boldsymbol{\theta} \approx \bar{\boldsymbol{\Psi}}\bar{\mathbf{y}}$$

where $\bar{\mathbf{y}}$ is obtained with the whitening procedure $\bar{\mathbf{y}} = \boldsymbol{\Lambda}_1^{-1/2}\mathbf{P}_1^T\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{\bar{m}})$ and $\bar{\boldsymbol{\Psi}}$ is given in (3.8), a KPI-related multiplicative fault is defined to be in the "process variable space" if

$$\bar{\mathbf{y}}_{mf} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\bar{\mathbf{y}}_{mf}}(\neq \mathbf{I}_{\bar{m}})\right) \ \& \ \bar{\boldsymbol{\Psi}}_f = \bar{\boldsymbol{\Psi}}$$

and in the "coefficient space" if

$$\bar{\boldsymbol{\Psi}}_f \neq \bar{\boldsymbol{\Psi}} \ \& \ \bar{\mathbf{y}}_{mf} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_{\bar{m}}\right)$$

where $\bar{\mathbf{y}}_{mf} = \boldsymbol{\Lambda}_1^{-1/2}\mathbf{P}_1^T\mathbf{y}_{mf}$ with $\mathbf{y}_{mf}$ denoting the normalized faulty process measurements. $\bar{\boldsymbol{\Psi}}_f$ is the faulty coefficient matrix defined as

$$\bar{\boldsymbol{\Psi}}_f = \frac{\boldsymbol{\Theta}_{mf}((\boldsymbol{\Lambda}_1\mathbf{K}_{f_1})^{-1/2}(\mathbf{P}_1^T + \boldsymbol{\Delta}\mathbf{P}_{f_1}^T)\mathbf{Y}_{mf})^T}{N_f - 1} \tag{6.6}$$

with

$$\begin{bmatrix}\mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_{f_1} & \mathbf{P}_2 + \boldsymbol{\Delta}\mathbf{P}_{f_2}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_1\mathbf{K}_{f_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_2\mathbf{K}_{f_2}\end{bmatrix} \times \begin{bmatrix}\mathbf{P}_1^T + \boldsymbol{\Delta}\mathbf{P}_{f_1}^T \\ \mathbf{P}_2^T + \boldsymbol{\Delta}\mathbf{P}_{f_2}^T\end{bmatrix} = \frac{\mathbf{Y}_{mf}\mathbf{Y}_{mf}^T}{N_f - 1},$$

$$\mathbf{K}_{f_1} = \ \text{diag}(\kappa_{f_1}, \cdots, \kappa_{f_{\bar{m}}}), \ \mathbf{K}_{f_2} = \text{diag}(\kappa_{f_{\bar{m}+1}}, \cdots, \kappa_{f_m})$$

be an SVD of $\frac{\mathbf{Y}_{mf}\mathbf{Y}_{mf}^T}{N_f - 1}$. $\boldsymbol{\Theta}_{mf} \in \mathcal{R}^{l \times N_f}$ and $\mathbf{Y}_{mf} \in \mathcal{R}^{m \times N_f}$ denote the normalized data matrices in the multiplicative faulty situation.

The objective of this chapter is to identify the most critical KPI-related/-unrelated process variable(s) that is(are) influenced by the multiplicative faults.

6.2 KPI-related multiplicative fault diagnosis

The aim of this section is to diagnose those KPI-related multiplicative faults in the "process variable space". Suppose that the multiplicative fault has influenced the correlation

matrices $\mathbf{\Sigma}_y$ and $\mathbf{\Sigma}_\theta$. Based on (6.5), the impact of multiplicative fault on the process variables is characterized as

$$\mathbf{\Sigma}_{y_{mf}} := E(\mathbf{y}_{mf}\mathbf{y}_{mf}^T) = \mathbf{F}\mathbf{\Sigma}_y\mathbf{F}^T \tag{6.7}$$

where

$$\mathbf{\Sigma}_y \approx \frac{\mathbf{Y}\mathbf{Y}^T}{N-1},\ \mathbf{\Sigma}_{y_{mf}} \approx \frac{\mathbf{Y}_{mf}\mathbf{Y}_{mf}^T}{N_f-1} \tag{6.8}$$

denote the covariance matrices of the normal and faulty data, respectively. In this study, we only consider these changes that increase the variabilities of the process variables as faults and we assume $rank(\mathbf{F}) = m$. Those changes which decrease variabilities are supposed to have no negative impacts on the system performance. From (6.7) we know that the covariance matrices of the fault-free and faulty process variables are of the same rank.

Considering possible redundancies and collinearities, the multiplicative fault $\mathbf{F}$ can then be estimated as

$$\hat{\mathbf{F}} = \mathbf{V}_1\mathbf{M}_1^{1/2}\mathbf{\Lambda}_1^{-1/2}\mathbf{P}_1^T$$

where

$$\begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}\begin{bmatrix}\mathbf{M}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{M}_2\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^T\\ \mathbf{V}_2^T\end{bmatrix} = \mathbf{\Sigma}_{y_{mf}},$$
$$\mathbf{M}_1 = \text{ diag}(m_1^2,\cdots,m_{\bar{m}}^2),\ \mathbf{M}_2 = \text{diag}(m_{\bar{m}+1}^2,\cdots,m_n^2),$$
$$m_1^2 \geq \cdots \geq m_{\bar{m}}^2 >> m_{\bar{m}+1}^2 \geq \cdots \geq m_m^2 = 0$$

is an SVD of $\mathbf{\Sigma}_{y_{mf}}$, $\mathbf{P}_1$ and $\mathbf{\Lambda}_1$ are given in (3.7).

Although the estimated value $\hat{\mathbf{F}}$ plays an important role for fault analysis, in practice it is more convenient to directly find out the root cause of the undesired variations. For this purpose, it is necessary to identify $\mathbf{G}$ (instead of $\mathbf{F}$) satisfying $\mathbf{G} = \mathbf{F}^+$ as

$$\mathbf{G} = \mathbf{P}_1\mathbf{\Lambda}_1^{1/2}\mathbf{M}_1^{-1/2}\mathbf{V}_1^T := \begin{bmatrix}\mathbf{g}_1^T\\ \vdots\\ \mathbf{g}_m^T\end{bmatrix} \in \mathcal{R}^{m\times m}. \tag{6.9}$$

From (6.7), it is clear that

$$\hat{\mathbf{y}} = \mathbf{G}y_{mf}$$

denotes the fault-free process variables which are reconstructed from all the multiplicative faults. Nevertheless, our purpose is to calculate the contributions of multiplicative fault from each component to the test statistic. Denote

$$\mathbf{G}_i = \begin{bmatrix}\mathbf{g}_1 & \cdots & \mathbf{g}_{i-1} & \mathbf{\xi}_i & \mathbf{g}_{i+1} & \cdots & \mathbf{g}_m\end{bmatrix}^T \in \mathcal{R}^{m\times m} \tag{6.10}$$

where $\boldsymbol{\xi}_i$ is the i-th column of the identity matrix, then

$$\hat{\mathbf{y}}_{f_i} = \mathbf{G}_i \mathbf{y}_{mf}$$

contains the abnormal/undesired variation only in the i-th component. The unexpected covariances among the remaining variables as well as their variances have been reconstructed from the fault scenario. If there are multiple measurable variables corresponding to the same component, then ξ_i should be extended to a proper matrix as well. The contribution of the i-th component fault to the $T^2_{\hat{\theta}}$ statistic is

$$\Delta T^2_{\hat{\theta},i} = T^2_{\hat{\theta},mf} - T^2_{\hat{\theta},R}$$

where

$$T^2_{\hat{\theta},mf} = \hat{\mathbf{y}}^T_{f_i} \mathbf{P}_1 \boldsymbol{\Lambda}_1^{-1/2} \mathbf{S}_1 \mathbf{S}_1^T \boldsymbol{\Lambda}_1^{-1/2} \mathbf{P}_1^T \hat{\mathbf{y}}_{f_i}$$

is the calculated statistic in faulty scenario and

$$\begin{aligned} T^2_{\hat{\theta},R} = \hat{\mathbf{y}}^T_{f_i} ((\mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_1)(\boldsymbol{\Lambda}_1 \mathbf{K}_1)^{-1/2} \mathbf{S}_1 \\ \times \mathbf{S}_1^T (\boldsymbol{\Lambda}_1 \mathbf{K}_1)^{-1/2} (\mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_1)^T) \hat{\mathbf{y}}_{f_i} \end{aligned}$$

is the test statistic reconstructed from the i-th component fault with

$$\begin{aligned} \mathbf{G}_i \boldsymbol{\Sigma}_{y_{mf}} \mathbf{G}_i^T = \begin{bmatrix} \mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_1 & \mathbf{P}_2 + \boldsymbol{\Delta}\mathbf{P}_2 \end{bmatrix} \\ \times \begin{bmatrix} \boldsymbol{\Lambda}_1 \mathbf{K}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_2 \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^T + \boldsymbol{\Delta}\mathbf{P}_1^T \\ \mathbf{P}_2^T + \boldsymbol{\Delta}\mathbf{P}_2^T \end{bmatrix}, \\ \mathbf{K}_1 = \ \mathrm{diag}(\kappa_1, \cdots, \kappa_{\bar{m}}),\ \mathbf{K}_2 = \mathrm{diag}(\kappa_{\bar{m}+1}, \cdots, \kappa_m). \end{aligned} \tag{6.11}$$

As a result, the contribution of the i-th component fault to the $T^2_{\hat{\theta}}$ statistic is calculated as

$$\Delta T^2_{\hat{\theta},i} = \mathbf{y}^T_{mf} \mathbf{G}_i^T \boldsymbol{\Phi}_{\hat{\theta},i} \mathbf{G}_i \mathbf{y}_{mf} \tag{6.12}$$

where

$$\begin{aligned} \boldsymbol{\Phi}_{\hat{\theta},i} = & -\boldsymbol{\Delta}\mathbf{P}_1 (\boldsymbol{\Lambda}_1 \mathbf{K}_1)^{-1/2} \mathbf{S}_1 \mathbf{S}_1^T (\boldsymbol{\Lambda}_1 \mathbf{K}_1)^{-1/2} (\mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_1)^T \\ & + \mathbf{P}_1 \boldsymbol{\Pi}_{y,1} \mathbf{S}_1 \mathbf{S}_1^T (\boldsymbol{\Lambda}_1 \mathbf{K}_1)^{-1/2} (\mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_1)^T \\ & + \mathbf{P}_1 \boldsymbol{\Lambda}_1^{-1/2} \mathbf{S}_1 \mathbf{S}_1^T \boldsymbol{\Pi}_{y,1} (\mathbf{P}_1 + \boldsymbol{\Delta}\mathbf{P}_1)^T \\ & - \mathbf{P}_1 \boldsymbol{\Lambda}_1^{-1/2} \mathbf{S}_1 \mathbf{S}_1^T \boldsymbol{\Lambda}_1^{-1/2} \boldsymbol{\Delta}\mathbf{P}_1^T \end{aligned}$$

with

$$\mathbf{\Pi}_{y,1} = \text{diag}\left(\frac{\sqrt{\kappa_1}-1}{\lambda_1\sqrt{\kappa_1}}, \cdots, \frac{\sqrt{\kappa_{\bar{m}}}-1}{\lambda_{\bar{m}}\sqrt{\kappa_{\bar{m}}}}\right).$$

Repeat (6.10), (6.11) and (6.12) for $i = 1, \cdots, \gamma$ where $\gamma \leq m$ is number of involved components, contributions of each component fault to the $T^2_{\hat{\theta}}$ test statistic are obtained. By plotting $\Delta T^2_{\hat{\theta},i}$, for $i = 1, \cdots, \gamma$, or its sample average on one chart over a certain time window, the most critical components causing KPI degradation can be intuitively isolated by comparing the magnitudes.

6.3 KPI-unrelated multiplicative fault diagnosis

As mentioned in the previous sections, those KPI-unrelated multiplicative faults have a great influence on the performance of the whole process in the long run, especially for the components' service life. In order to find out the fault at an early stage, it is urgent to establish an automatic diagnosis tool in the residual subspace. The diagnosis procedure in the residual subspaces is quite similar to the one in the "process variable space" for KPI-related faults. Based on (6.11), compute the contribution of the i-th component fault to the $T^2_{\theta^\perp}$ statistic as

$$\Delta T^2_{\theta^\perp,i} = \mathbf{y}^T_{mf}\mathbf{G}^T_i\mathbf{\Phi}_{\theta^\perp,i}\mathbf{G}_i\mathbf{y}_{mf} \tag{6.13}$$

where

$$\begin{aligned}
\mathbf{\Phi}_{\theta^\perp,i} = &-\lambda^2_m\mathbf{\Delta P}_1(\mathbf{\Lambda}_1\mathbf{K}_1)^{-1/2}\mathbf{S}_2\mathbf{S}^T_2(\mathbf{\Lambda}_1\mathbf{K}_1)^{-1/2} \\
&\times(\mathbf{P}_1+\mathbf{\Delta P}_1)^T + \lambda^2_m\mathbf{P}_1\mathbf{\Pi}_{y,1}\mathbf{S}_2\mathbf{S}^T_2(\mathbf{\Lambda}_1\mathbf{K}_1)^{-1/2}(\mathbf{P}_1+\mathbf{\Delta P}_1)^T \\
&+\lambda^2_m\mathbf{P}_1\mathbf{\Lambda}^{-1/2}_1\mathbf{S}_2\mathbf{S}^T_2\mathbf{\Pi}_{y,1}(\mathbf{P}_1+\mathbf{\Delta P}_1)^T \\
&-\lambda^2_m\mathbf{P}_1\mathbf{\Lambda}^{-1/2}_1\mathbf{S}_2\mathbf{S}^T_2\mathbf{\Lambda}^{-1/2}_1\mathbf{\Delta P}^T_1 \\
&-\mathbf{\Delta P}_2(\mathbf{\Xi}_2-\mathbf{\Pi}_{y,2})(\mathbf{P}_2+\mathbf{\Delta P}_2)^T \\
&+\mathbf{P}_2\,\mathbf{\Pi}_{y,2}(\mathbf{P}_2+\mathbf{\Delta P}_2)^T - \mathbf{P}_2\mathbf{\Xi}\mathbf{\Delta P}^T_2
\end{aligned}$$

with

$$\mathbf{\Pi}_{y,2} = \text{diag}\left(\frac{\lambda^2_m(\kappa_{\bar{m}+1}-1)}{\lambda^2_{\bar{m}+1}\kappa_{\bar{m}+1}}, \cdots, \frac{\kappa_m-1}{\kappa_m}\right),\ \mathbf{\Xi} = \text{diag}(\frac{\lambda^2_m}{\lambda^2_{\bar{m}+1}}, \cdots, \frac{\lambda^2_m}{\lambda^2_{m-1}}, 1).$$

By plotting $\Delta T^2_{\theta^\perp,i}$ for $i = 1, \cdots, \gamma$, on one chart, we can identify the most critical components suffering KPI-unrelated fault.

6.4 Thresholds for multiplicative fault diagnosis

Sometimes comparing the magnitudes of $\Delta T^2_{\hat{\theta},i}$, for $i = 1, \cdots, \gamma$, or $\Delta T^2_{\theta^\perp,i}$, for $i = 1, \cdots, \gamma$, might give misleading results since even in the fault-free case, the contributions of all the components to the performance indices are uneven. Thus, similar to the well-established process monitoring techniques, it is necessary to set some threshold to increase fault diagnosis performance. Many statistical tools exist for this purpose. In the field of process monitoring and fault diagnosis, Box's theorems on quadratic forms for variance analysis [11] and the kernel density estimation [102] (KDE) based technique are two preferred tools.

Box's theorems based approach. Under the assumption that $\mathbf{y}$ is multivariate normal distributed, using Box's theorems given in [11], the threshold for (6.12) can be determined as

$$J_{th,\Delta T^2_{\hat{\theta},i}} = g_{\hat{\theta},i}\chi^2_{1-\alpha}(h_{\hat{\theta},i}), \forall i = 1, \cdots, \gamma \tag{6.14}$$

where

$$g_{\hat{\theta},i} = \frac{\sigma^2_{\hat{\theta},i}}{2\mu_{\hat{\theta},i}},\ h_{\hat{\theta},i} = \frac{2\mu^2_{\hat{\theta},i}}{\sigma^2_{\hat{\theta},i}}$$

with

$$\mu_{\hat{\theta},i} = \frac{1}{N}\sum_{k=1}^{N}\Delta T^2_{\hat{\theta},i}(k),\ \sigma^2_{\hat{\theta},i} = \frac{1}{N-1}\sum_{k=1}^{N}\left(\Delta T^2_{\hat{\theta},i}(k) - \mu_{\hat{\theta},i}\right)^2.$$

Note that the above thresholds are for an individual sample. In our approach, the features of multiplicative fault are identified from a piece of online data which are sufficient for covariance matrix estimation. Thus under the assumption that $\Delta T^2_{\hat{\theta},i}(k), k = 1, \cdots, N_f$, is identically independent distributed with N_f denoting the size of the online data, the following thresholds are obtained

$$J^{N_f}_{th,\Delta T^2_{\hat{\theta},i}} = \frac{g_{\hat{\theta},i}}{N_f}\chi^2_{1-\alpha}(N_f h_{\hat{\theta},i}), i = 1, \cdots, \gamma \tag{6.15}$$

for testing $\frac{1}{N_f}\sum_{k=1}^{N_f}\Delta T^2_{\hat{\theta},i}, i = 1, \cdots, \gamma$.

In the same way, the thresholds for (6.13) are given as

$$J^{N_f}_{th,\Delta T^2_{\theta^\perp,i}} = \frac{g_{\theta^\perp,i}}{N_f}\chi^2_{1-\alpha}(N_f h_{\theta^\perp,i}), i = 1, \cdots, \gamma \tag{6.16}$$

where

$$g_{\theta^\perp,i} = \frac{\sigma^2_{\theta^\perp,i}}{2\mu_{\theta^\perp,i}},\ h_{\theta^\perp,i} = \frac{2\mu^2_{\theta^\perp,i}}{\sigma^2_{\theta^\perp,i}},$$

$$\mu_{\theta^\perp,i} = \frac{1}{N}\sum_{k=1}^{N}\Delta T^2_{\theta^\perp,i}(k),\ \sigma^2_{\theta^\perp,i} = \frac{1}{N-1}\sum_{k=1}^{N}\left(\Delta T^2_{\theta^\perp,i}(k) - \mu_{\theta^\perp,i}\right)^2.$$

Algorithm 6.1. *KPI-based multiplicative fault diagnosis*

S1: *Run Algorithm 3.2.*

S2 *Determine diagnosis thresholds from (6.15), (6.16) or using the KDE method.*

Once a multiplicative fault is detected, based on the faulty data $\mathbf{Y}_{mf} \in \mathcal{R}^{m \times N_f}$,

S3: *Estimate* $\mathbf{\Sigma}_{y_{mf}}$ *from (6.8) and build* $\mathbf{G}_i$ *as (6.9) and (6.10) for* $i = 1, \cdots, \gamma$.

S4 *For each* i, *do an SVD on* $\mathbf{G}_i \mathbf{\Sigma}_{y_{mf}} \mathbf{G}_i^T \Longrightarrow \mathbf{\Delta P}_1, \mathbf{K}_1; \mathbf{\Delta P}_2, \mathbf{K}_2$.

S5 *Build* $\Delta T^2_{\hat{\theta},i}$ *or* $\Delta T^2_{\theta^\perp,i}$ *for* $i = 1, \cdots, \gamma$.

S6 *Check the decision logic:*

$$\begin{cases} \Delta T^2_{\hat{\theta},i} > J^{N_f}_{th,\Delta T^2_{\hat{\theta},i}} & \Rightarrow \textit{the } i\textit{-th variable is responsable for KPI-related performance degradation} \\ \Delta T^2_{\theta^\perp,i} > J^{N_f}_{th,\Delta T^2_{\theta^\perp,i}} & \Rightarrow \textit{the } i\textit{-th variable is responsable for KPI-unrealted performance degradation.} \end{cases}$$

KDE-based approach. KDE is a non-parametric approach to estimate the probability density function of a random variable. It is a fundamental data smoothing problem where inferences about the population are made from finite data samples. Different from the previous approach, from application viewpoint, KDE does not require that the process variables $\mathbf{y}$ should be normal distributed. Based on the normal process data, the $\Delta T^2_{\hat{\theta},i}(k)$, for $i = 1, \cdots, \gamma$, and $\Delta T^2_{\theta^\perp,i}(k)$, for $i = 1, \cdots, \gamma$,are firstly computed for $k = 1, \cdots, N$. Then the empirical density estimates of them are obtained by means of kernel extraction, whose principle is quite similar to the histogram. At last, the thresholds are determined for given significance level α. In Matlab, for instance, the "ksdensity" command can be used for it.

It is important to note that the methods for determining the thresholds for fault diagnosis in this subsection are different from the one discussed in Chapter 3 for process monitoring. Box's theorems based approach is a parametric one, it utilizes the *a priori* distribution information, i.e χ^2. Since the test statistics $\Delta T^2_{\hat{\theta},i}(k)$, for $i = 1, \cdots, \gamma$, and $\Delta T^2_{\theta^\perp,i}(k)$, for $i = 1, \cdots, \gamma$, are not normalized, the impacts of their mean and variance must be considered. When setting the thresholds, the mean and variance are estimated from the offline training data of limited size. Thus the quality of training data plays an important role (should be sufficiently excited). The KDE-based approach however, requires no *a priori* distribution knowledge. The distribution structure as well the associated parameters are estimated from the training data. Thus the quality of training data plays an even influential role. In this chapter, it is assumed that sufficient informative data is available thus modelling error is neglected. Nevertheless, to deal with this issue, adaptive implementation of the proposed method is of great interest.

Remarks on application scope and performance of the proposed scheme: The method

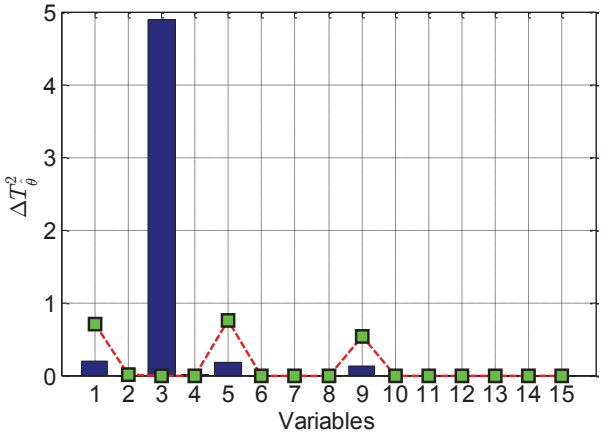


Figure 6.1: Diagnosis result when fault happens in the "process variable space" with the KPIs influenced

presented in this chapter is developed for linear (Gaussian) steady processes. It is a data-driven approach applicable for general automation industrial processes. In practical applications, process nonlinearity, time-varying parameters and strong dynamics could decrease the diagnosis performance. Thus multi-mode, adaptive and dynamic diagnosis approaches of multiplicative fault are of practical interest and demand more research activities.

The design procedures for the KPI-based multiplicative fault detection and diagnosis are summarized in Algorithm 6.1.

6.5 Numerical examples

In this section, the same numerical model given in Chapter 3 is considered. In the following, we will show two examples for the faults that occur in the "process variable space" with and without KPIs being influenced, respectively. The first fault is injected into the 3rd variable with $F(3,6) = 5$. It influences the variances of both KPIs and has been detected by the $T^2_{\hat{\theta}}$ test statistic. Figure 6.1 gives the diagnosis result. We can see that the contribution of the 3rd variable is the largest and crosses the threshold. It indicates that the real root cause is identified.

For the next example, we set $F(11,6) = 5$, it is injected into the 11th variable. This fault does not influence any KPI. However, the monitoring results indicate that it is a KPI-unrelated fault since the $T^2_{\theta^\perp}$ statistic has detected the fault. Figure 6.2 shows the diagnosis result, from which we can see the 11th variable is correctly identified as the root cause.

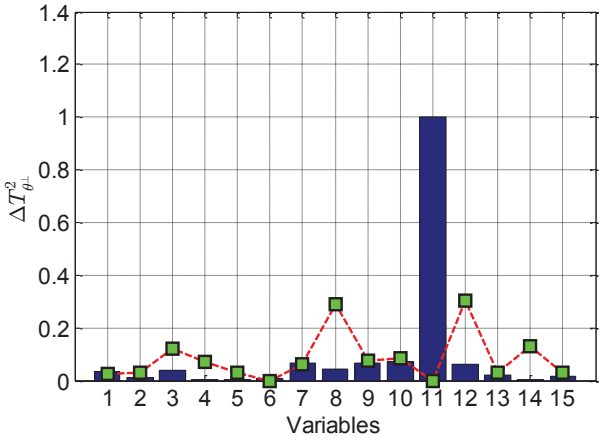


Figure 6.2: Diagnosis result when fault happens in the "process variable space" with the KPIs not influenced

In addition, monte-carlo-simulation can be found in our recently work [47].

6.6 Concluding remarks

In this chapter, a new approach for the diagnosis of multiplicative performance degradation is discussed. It is a data-driven method and requires less engineering effort. Different from the RBC-based approach which aims at diagnosing mean value change, the proposed approach focuses on diagnosing covariance matrix change, which is quite common in industry but not yet well addressed. The advantage of this approach is that no *a prior* information about the fault is required. It makes use of the parameters extracted from the fault-free data during the offline training phase and the online faulty data. The major objective is to increase the OEE by narrowing down the investigation scope for the operators. In addition, the diagnosis results can provide valuable guidelines for selecting proper corrective actions.

7 Application to benchmark processes

In this chapter, we will demonstrate the applications of the proposed methods to realistic benchmark processes. Depending on the application scope of different methods, three benchmark processes are used. The methods developed in Chapters 3 and 6, which are suitable for large-scale, static automation processes, are applied to the Tennessee Eastman (TE) benchmark process [30]. The continuous stirred tank heater (CSTH) benchmark [107], which is a common subsystem widely existing in the chemical industry, is used to test the dynamic methods proposed in Chapter 4. Finally, the methods proposed in Chapter 5 are applied to the paper drying (PD) benchmark process [8, 9], which consists of many spatially distributed heating cylinders and is the most important section in a paper machine.

7.1 Case studies on the TE benchmark process

7.1.1 Process description

The TE benchmark is developed based on an actual industrial process by the corporate process control group of the Eastman Chemical Company. It is part of a complex chemical plant where the reactants are provided by other upstream production facilities and the products are further separated in a downstream refining section. Although some components like kinetics, process and operating conditions are modified to protect the proprietary nature of the process, the benchmark is highly realistic and widely used to test various process control and monitoring schemes. Figure 7.1 shows the diagram of the process. Four gaseous reactants A, D, E and C are fed to the reactor and the following reactions take place

$$\begin{aligned} A(g) + C(g) + D(g) &\rightarrow G(liq), \\ A(g) + C(g) + E(g) &\rightarrow H(liq), \\ A(g) + E(g) &\rightarrow F(liq), \\ 3D(g) &\rightarrow 2F(liq) \end{aligned}$$

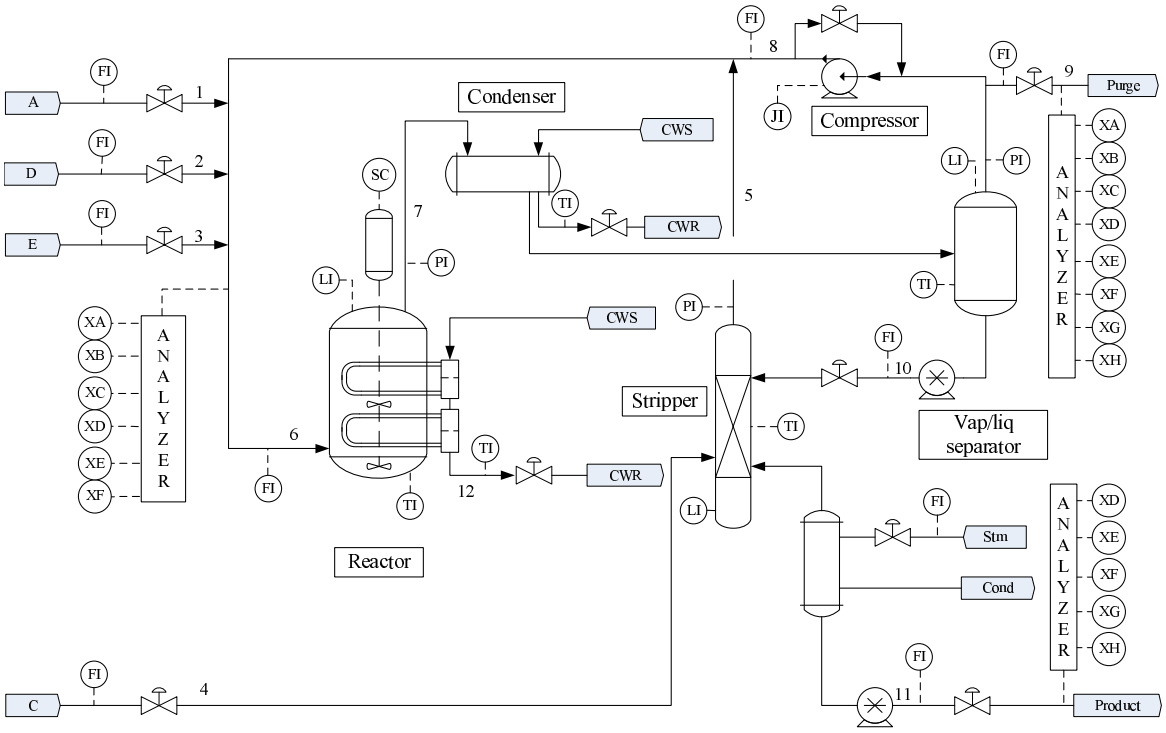


Figure 7.1: The Tennessee Eastman test problem [30]

where G and H are liquid products and F is a byproduct. Since all reactions are exothermic, a water-cooling system is used to transfer additional heat in the reactor. The generated product stream passes through a condenser where liquid products are produced. Then, the product stream is further fed to a "Vap/liq" separator where the noncondensed components are recycled back to the reactor and the condensed components move to a stripper which removes the remaining reactants. Finally, the liquid products exit the stripper base and are separated in the downstream refining section. The automation degree is very high. As listed in Table 7.1, twenty-two process variables are continuously measured. There are 12 manipulated variables, which are more than necessary for controlling the process. In addition, three analyzers are available for measuring the amount of each component for stream 6, 9 and 11. The analyzers for stream 6 and 9 need 6 minutes to complete the analysis and the analyzer for stream 11 takes 15 minutes. As a result, the minimal sampling time for streams 6 and 9 are 6 minutes and for stream 11 is 15 minutes. In this section, the MATLAB/Simulink programs provided by Ricker is used, which is available at the website

http://depts.washington.edu/control/LARRY/TE/download.html

and can be downloaded. For our study, the process is running under the mode number 1, *i.e.* G/H mass ratio is 50/50 and the production rate for G and H are both 7308 $kg\ h^{-1}$

Table 7.1: TE: Process measurements and manipulated variables

Block	Description	XMEAS No.	XMV No.
Feeds & Reactor	A feed (stream 1)	XMEAS(1)	XMV(3)
	D feed (stream 2)	XMEAS(2)	XMV(1)
	E feed (stream 3)	XMEAS(3)	XMV(2)
	A and C feed (stream 4)	XMEAS(4)	XMV(4)
	Compressor work	XMEAS(20)	
	Compressor recycle valve		XMV(5)
	Recycle flow (stream 8)	XMEAS(5)	
	Reactor feed rate (stream 6)	XMEAS(6)	
	Reactor pressure	XMEAS(7)	
	Reactor level	XMEAS(8)	
	Reactor temperature	XMEAS(9)	
	Reactor cooling water outlet temperature	XMEAS(21)	
	Reactor cooling water flow		XMV(10)
	Agitator speed		XMV(12)
Condenser & Separator	Separator temperature	XMEAS(11)	
	Separator level	XMEAS(12)	
	Separator pressure	XMEAS(13)	
	Separator underflow (stream 10)	XMEAS(14)	XMV(7)
	Condenser cooling water outlet temperature	XMEAS(22)	
	Condenser cooling water flow		XMV(11)
	Purge rate (stream 9)	XMEAS(10)	
	Purge valve (stream 9)		XMV(6)
Stripper	Stripper level	XMEAS(15)	
	Stripper pressure	XMEAS(16)	
	Stripper underflow (stream 11)	XMEAS(17)	XMV(8)
	Stripper temperature	XMEAS(18)	
	Stripper steam flow	XMEAS(19)	XMV(9)

[30], with the decentralized control strategy described in [97]. The control method is able to reject all the 20 disturbance defined in [30] except for the 6th, the 8th and the 13th[1], which makes it quite challenging to design a monitoring system. To this end, we take the 22 process variables and 9 manipulated variables (XMV(5), XMV(9) and XMV(12) are removed since they are constant under the control strategy and contain no useful information for monitoring and diagnosis) as the low-level process variables and the sampling time is set to be 36 seconds. The KPI is defined to be the operation cost as

$$\text{KPI (operation cost)} = \text{purge cost} + \text{product stream cost} + \text{compressor cost}.$$

Since the price for the components are different and the analysis results for purge flow (stream 9) and product flow (stream 11) are delayed by 6 and 15 minutes, respectively, the total operation cost is not always online available. In the following, we will apply the approaches proposed in Chapters 3 and 6 in such a realistic environment for two fault episodes.

[1]The 6th disturbance is "A feed loss", the process is shut down due to low stripper level; for the 8th and 13th disturbance, the process is stabilized but the output variation is quite large.

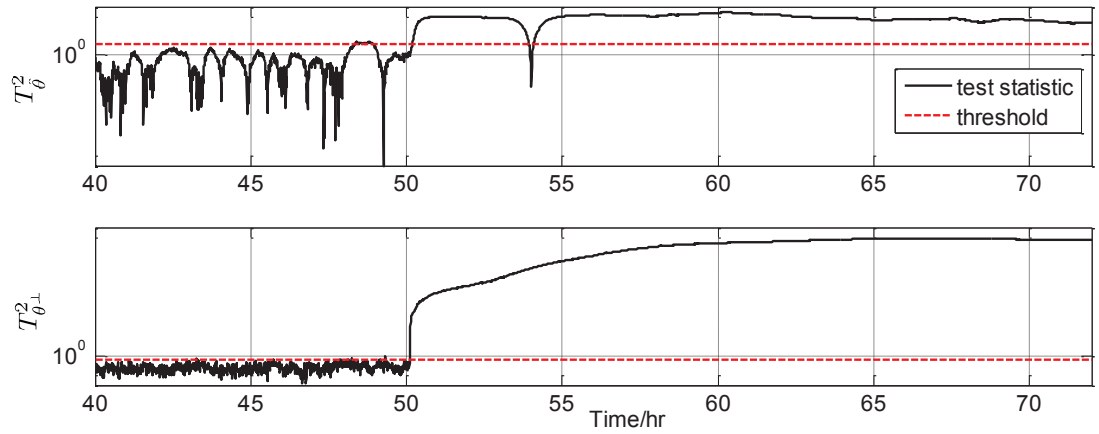


Figure 7.2: TE: Monitoring results for the additive fault episode

7.1.2 Detection of an additive fault

For the TE process, very little holdup is available for the stream 4, components A and C. As a result, flow variability of this feed stream is of particular concern. Motivated by it, we define an additive fault episode by reducing the setpoint for stream 4 by 23.5%. The Algorithm 3.2 is applied. Figure 7.2 shows the monitoring results for this fault episode. The total simulation time is 72 hours. The fault is injected from the 50th hour. For offline training purpose, the data collected for the first 48 hours are used. The significance level is defined to be 1%. From the plots we can see that both test statistics have detected the fault. The upper plot indicates that this fault is related to the KPI, *i.e.* KPI degradation occurs after the 50th hour. The lower plot shows that this fault causes KPI-unrelated performance degradation after the 50th hour as well. To validate the monitoring results, we plot the KPI and some selected process variables in Figure 7.3. We can see that the operation cost has increased by around 100 dollars in the steady state. The fault firstly causes a reduction in A/C feed. Under the decentralised control strategy, the compressor work is then reduced, which brings less recycle flow back to the reactor. To enhance the reactions, the D and E feeds are increased (A feed is almost unchanged). The purge rate is creased during the 55-65 hours. Since the D feed is the most expensive one, the total operation costs have increased. In addition, we can observe from Figure 7.3 that the amount of product H is reduced (H in product is not included in the KPI definition). It means that this fault causes KPI-unrelated performance degradation, which is in accordance with the monitoring result.

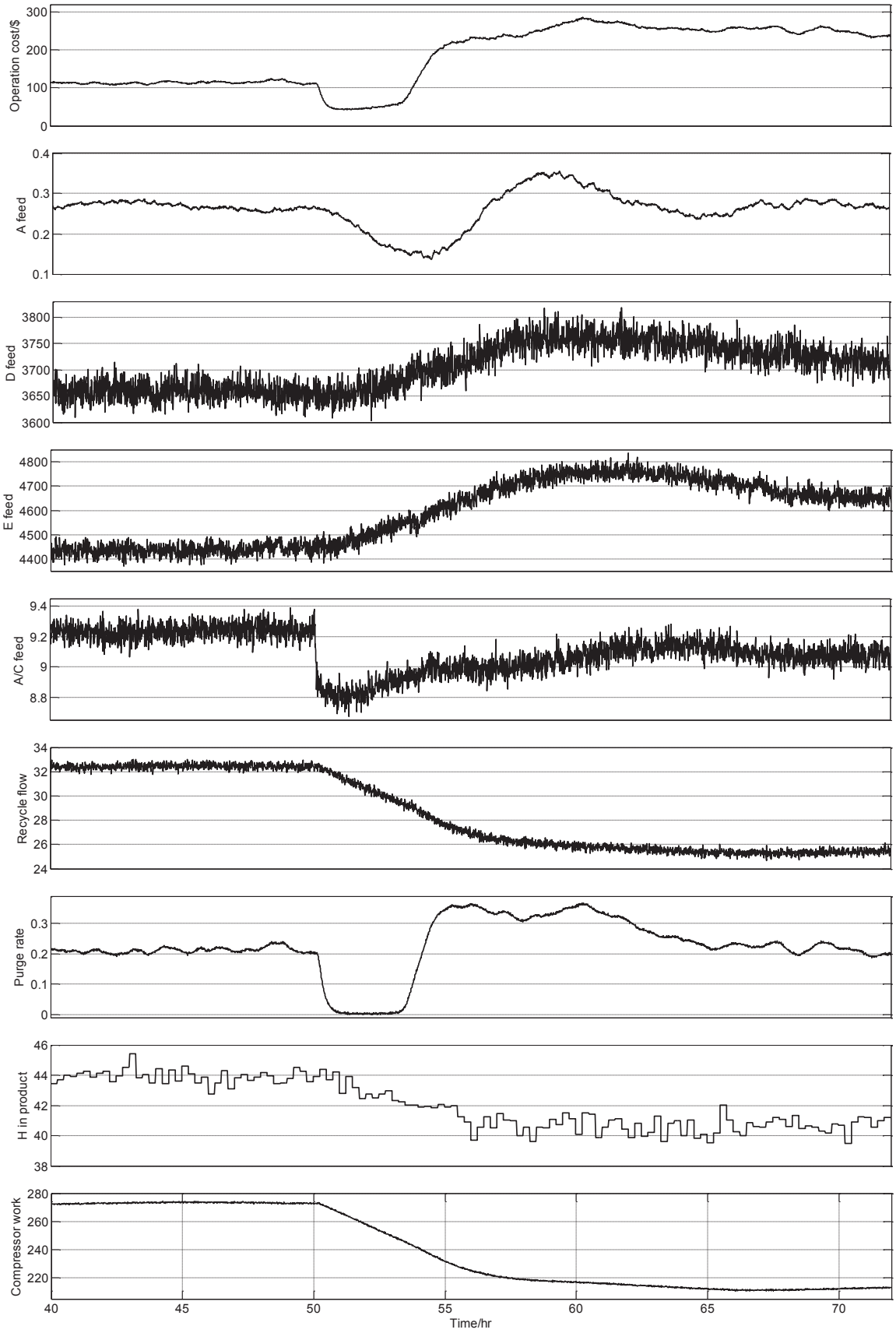
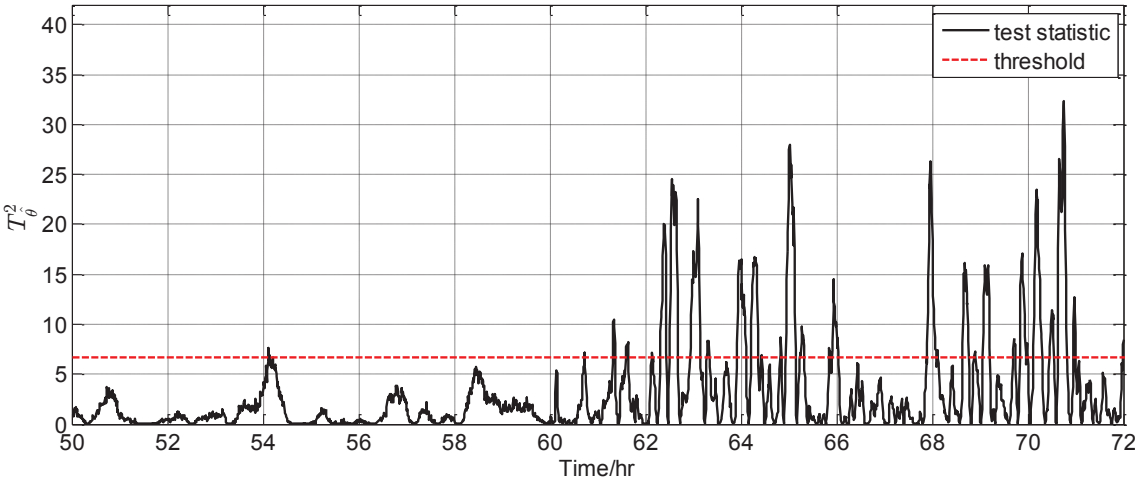
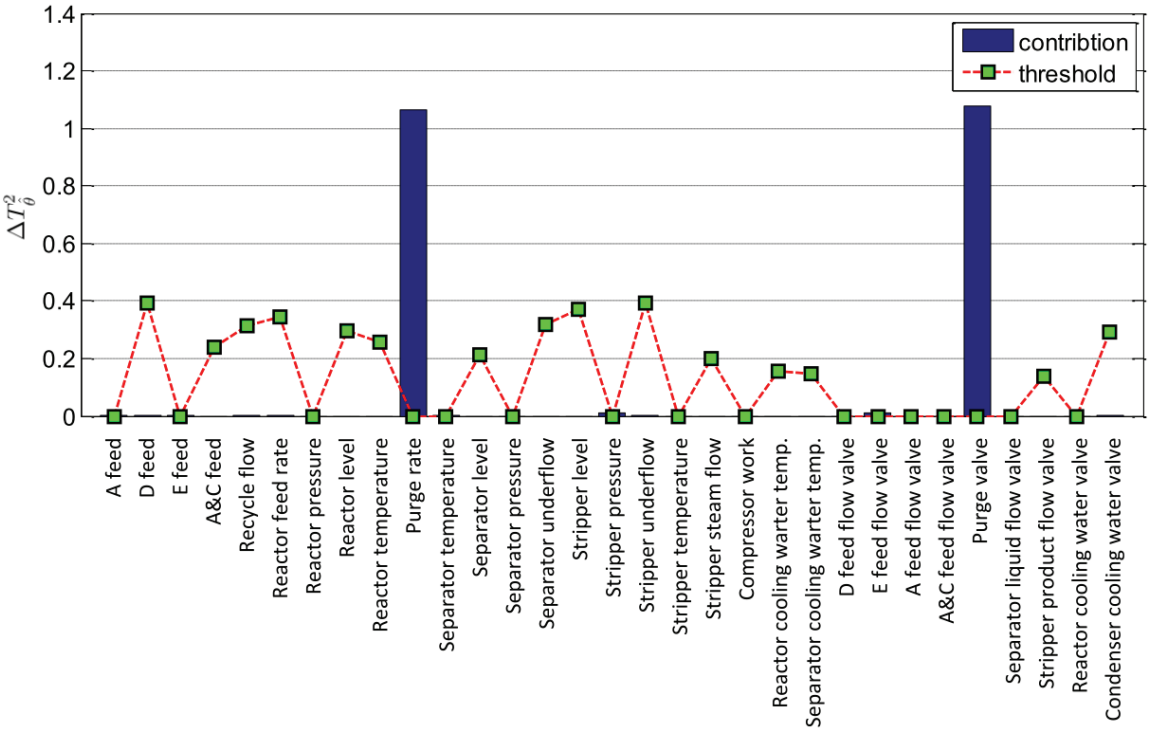


Figure 7.3: TE: KPI and selected low-level process variables

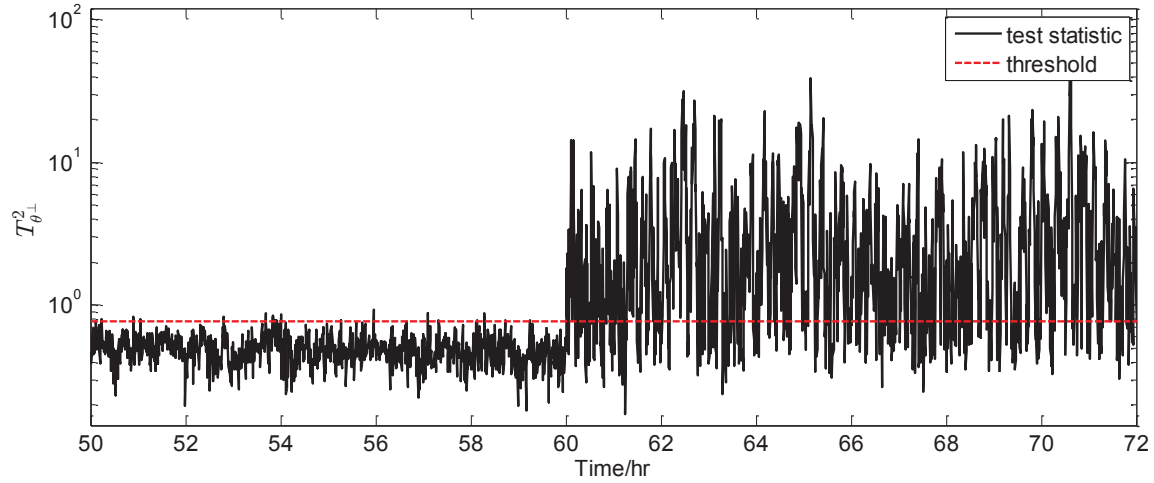


(a) Monitoring result on the KPI-related measurement subspace. At the 60th hour, a multiplicative fault happens.

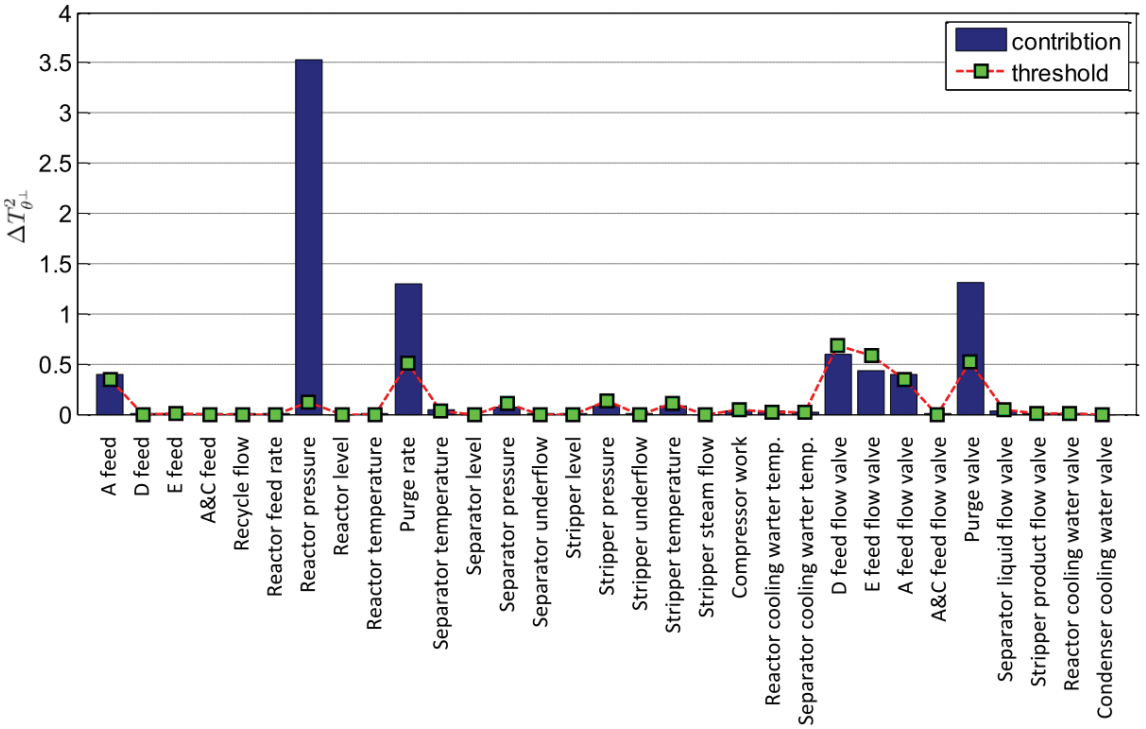


(b) Diagnosis result for KPI-related (multiplicative) performance degradation

Figure 7.4: TE: Monitoring and diagnosis results related to the KPI



(a) Monitoring result on the KPI-unrelated measurement subspace. At the 60th hour, a multiplicative fault happens.



(b) Diagnosis result for KPI-unrelated (multiplicative) performance degradation

Figure 7.5: TE: Monitoring and diagnosis results unrelated to the KPI

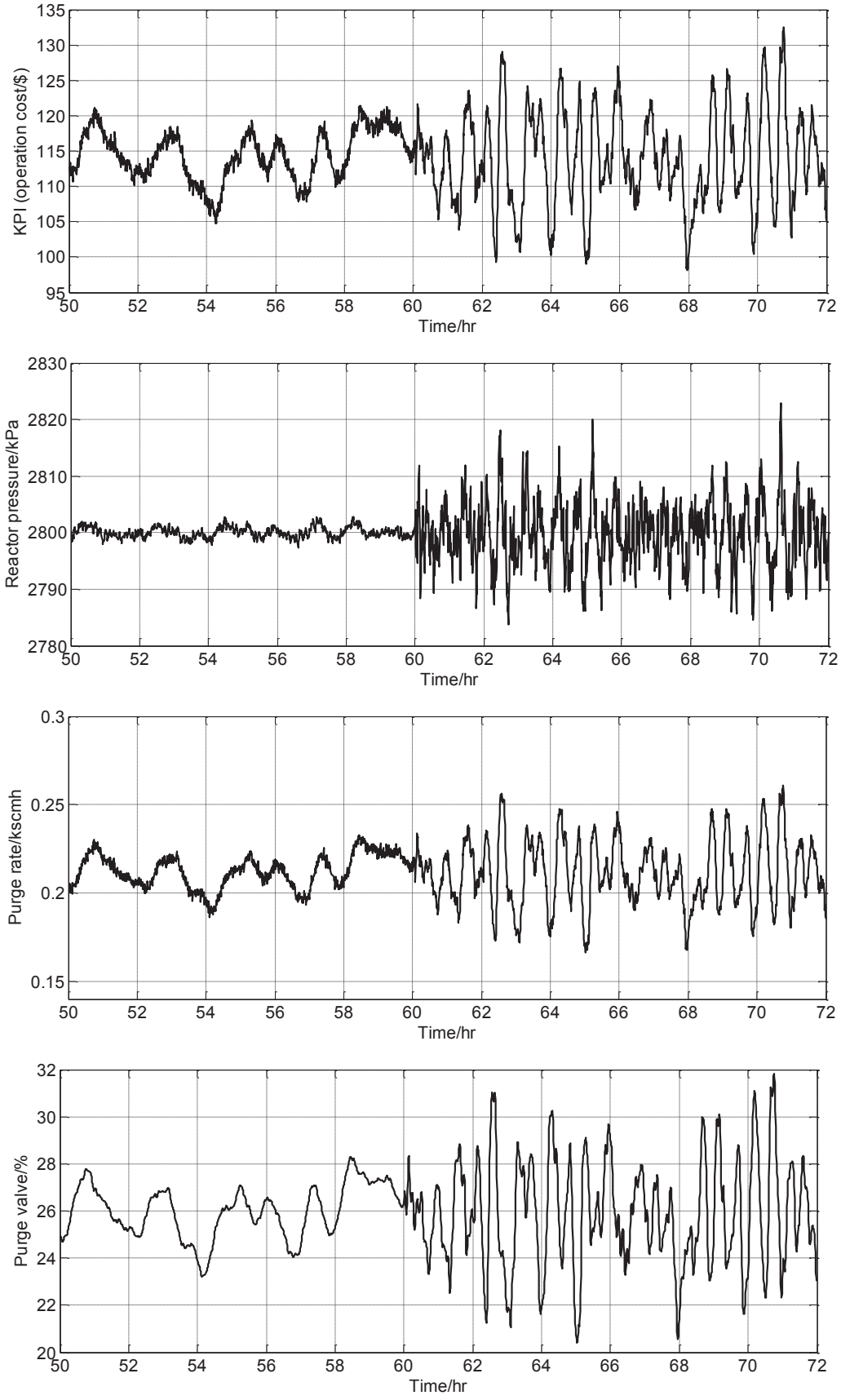


Figure 7.6: TE: KPI and selected low-level process variables

7.1.3 Detection and diagnosis of a multiplicative fault

The objective of this case study is to show the effectiveness of Algorithm 6.1 proposed in Chapter 6. We define a multiplicative fault episode by increasing the noise level of the reactor's pressure sensor from the 60th hour (the mean value is unchanged). The other simulation conditions are kept unchanged. Figure 7.4 shows the KPI-related monitoring and diagnosis results. We can see from Figure 7.4-a that the test statistic starts to raise alarms from the 61st hour. These intermittent alarms are much more frequent than false alarms and reflects the occurrence of a multiplicative fault. Figure 7.4-b gives the diagnosis result. It shows that "Purge rate" and "Purge value" are responsible for the KPI degradation. Simultaneously, Figure 7.5-a shows that KPI-unrelated performance degradation occurs. Its diagnosis result is given in Figure 7.5-b, from which we can see that "reactor pressure", "purge rate" and "purge valve" are responsible for it. To validate the achieved results, we plot the KPI and infected process variables in Figure 7.6. From it we can see that the variability of the operation cost is increased after the 60th hour. The noise level of the pressure sensor is significantly increased and its effect cannot be reduced by the controllers. Since the pressure measurements are used to generate the reference signal for the "purge rate" control loop, the variabilities of both "purge rate" and "purge valve" are increased. As "purge flow" is contributing to the total operation cost, it is directly related to the KPI. On the other hand, the reactor pressure is not included in the definition of KPI. However, tank pressure is very important for safe operation, which is a kind of KPI-unrelated performance for this case study.

7.2 Case studies on the CSTH benchmark process

7.2.1 Process description

The CSTH process is a common subsystem widely used in the chemical industry. It can be used to keep optimal temperature for reactants, water, reactions, *etc.*. In this section, we will use the simulation model provided by Thornhill, which is available at the website

http://personal-pages.ps.ic.ac.uk/∼nina/CSTHSimulation/index.htm

and can be downloaded. It is developed based on a pilot plant at the University of Alberta using first principles. As shown in Figure 7.7, hot and cold water are mixed first, where the hot water (HW) boiler is heated by the university campus steam supply. The mixture is then heated using the steam from the same central campus source through a heating coil. Finally, the heated water is drained from the tank through a long pipe. The simulation model is highly realistic. On the one hand, instrument, actuator and process nonlinearities have been carefully measured and taken into account in the model;

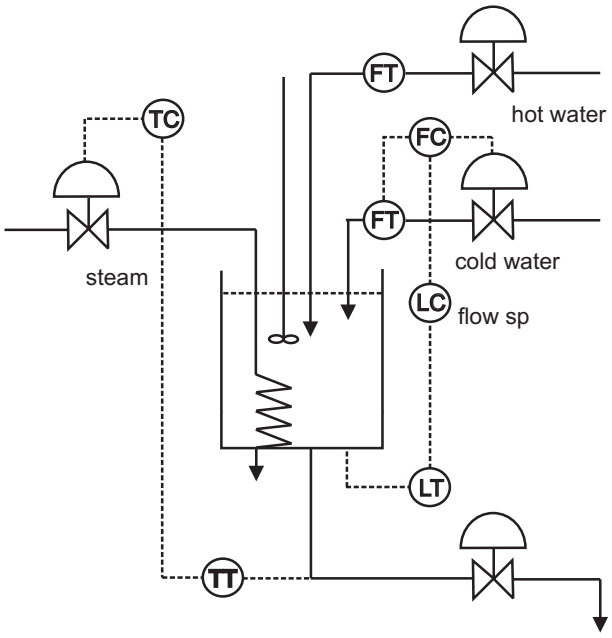


Figure 7.7: The continuous stirred tank heater [107]

Table 7.2: CSTH: Manipulated variables and process measurements

Block	Description	Unit
Process measurements	Mixed water temperature	$^{\circ}C$
	Water level	cm
	CW flow rate	liter/min
Manipulated variables	Steam valve	kJ/sec
	CW valve	liter/min

on the other hand, measured noise and disturbances are used. The CSTH is an automatic process. As listed in Table 7.2, there are three measurable process variables and two manipulated variables. The steam and cold water (CW) valves are controlled by proportional-integral (PI) controllers. Since cold water flow, (tank) water level and temperature are of main concern for a CSTH plant and measured for this configuration, we consider the three process measurements as KPIs. The utilities of the CSTH are shared service and subject to disturbances from other users, the mean values of process variables are therefore continuously changing. As a result, it is not preferable to use the static methods discussed previously to monitor its performance. In the following, we will apply the Algorithm 4.2 from Chapter 4 to detect four typical fault episodes.

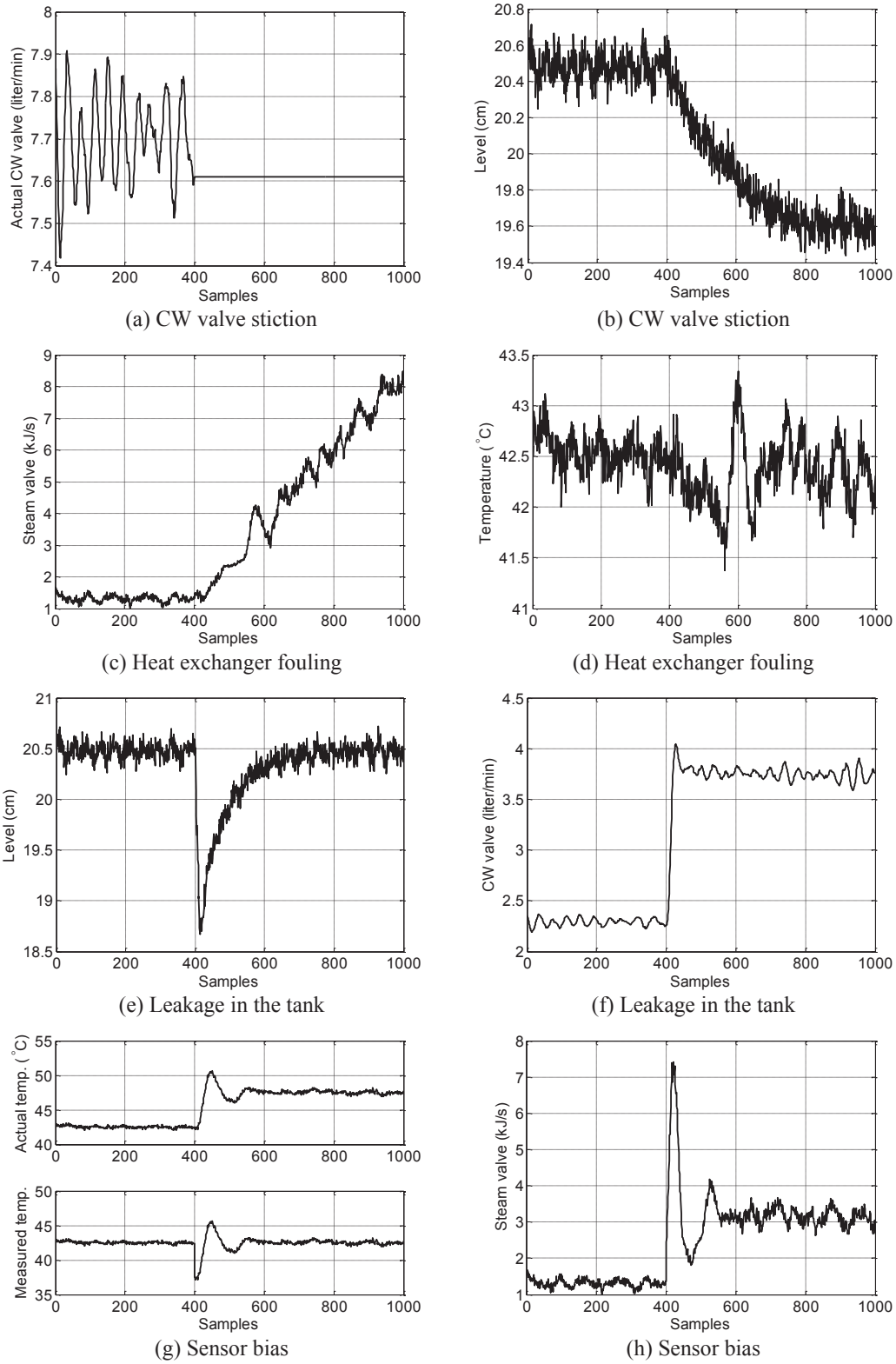


Figure 7.8: CSTH: Description of the fault episodes

7.2.2 Detection of four typical fault episodes

For our case studies, we set the following operating conditions

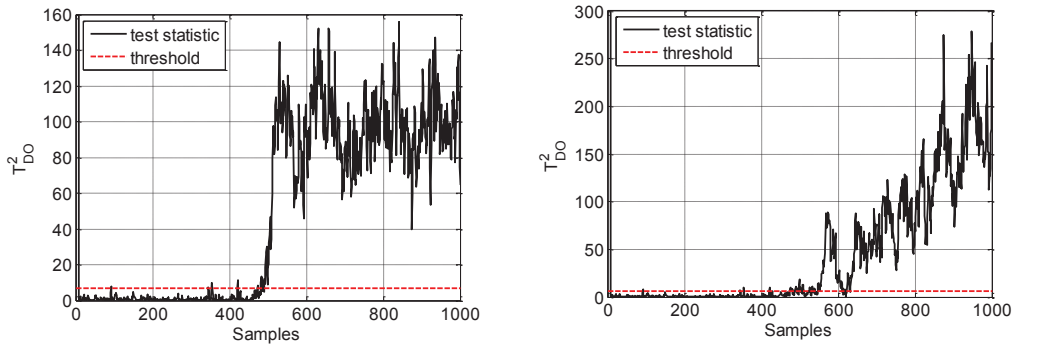
- *Temperature set point: 42.52 °C,*
- *Level set point: 20.48 cm,*
- *HW valve: 3.129 liter/min,*
- *HW temperature: 50 °C,*
- *CW temperature: 24 °C,*
- *Manual outflow valve: 50%.*

Four typical fault episodes originally defined in [43] are used:

- *CW valve stiction:* Valves are the most widely used actuators in the chemical industry. They have direct contact with various reactant and product streams and are frequently subject to malfunctions. Among them, stiction is a very common one. Figure 7.8-a shows the actual CW valve position subject to stiction, it is sticked at 7.609 $liter/min$ from the 400th sample. Since the controller's command is not executed and the sticked position is lower than the average, the water level decreases from the 400th sample, which can be seen from 7.8-b. Detection of this fault is urgent, as the water holdup is reduced and the tank will be empty.

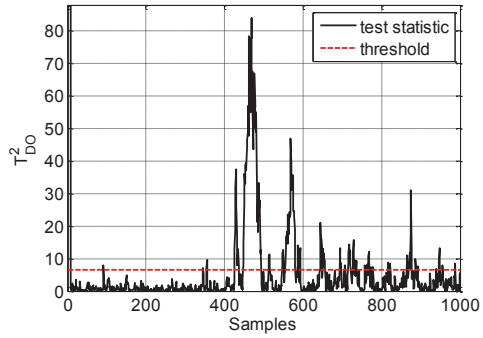
- *Heat exchanger fouling:* In practice, there is chalk in the water and it deposits gradually on the surface of the heating coil. As a result, the heat transfer efficiency from the coil to the water is reduced. Here we use a ramp function starting from the 400th sample to simulate this fault episode. As shown in Figure 7.8-c-d, in order to keep the desired temperature, the controller increases the openness of the steam valve. Although this fault is not as critical as the previous one, its detection is very important since the operation cost is increased.

- *Leakage in the tank:* This kind of fault happens more in old plants. It cannot only increase the operation cost, but also causes environmental pollution or even disasters. To show the effectiveness of the proposed methods for it, we simulate a hole at the bottom of the tank from the 400th sample. It is an abrupt fault. As can be seen from Figure 7.8-e-f, it first decreases the water level. In order to keep the desired level, the controller decides to increase openness of the CW valve.

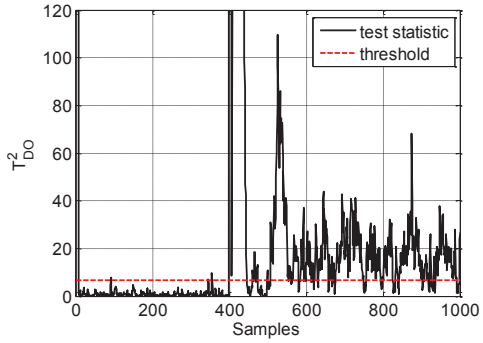


(a) Monitoring result for cold water valve stiction: constant

(b) Monitoring result for heat exchanger fouling: ramp



(c) Monitoring result for leakage in the stirred tank: step



(d) Monitoring result for temperature sensor bias: step

Figure 7.9: CSTH: KPI monitoring results

- *Temperature sensor bias:* Badly calibrated sensors exist in the practice. We simulate a bias of $-5\ ^{\circ}C$ for the temperature sensor from the 400th sample. Figure 7.8-g shows the actual and measured temperature of the water. Due to the feedback control action, the "measured signal" can be kept at the set point. However, the actual one is $5\ ^{\circ}C$ higher. It could be quite dangerous in practice, since the actual temperature may exceed the upper limit without letting the operator know it. In addition, to arrive at this unexpected higher temperature, the consumption of steam is increased, which can be seen from Figure 7.8-h.

To identify the parameters involved in the diagnostic observer, we have run the simulation program in the fault free case and collected 1600 samples of data. The sampling time is 1 second. The design parameters are chosen as: $s = 8$ and $s_p = 10$. The significance level α for determine the threshold is set to be 1%. Figure 7.9 shows the monitoring results for these four fault episodes. From it we can see that all fault episodes have been successfully detected.

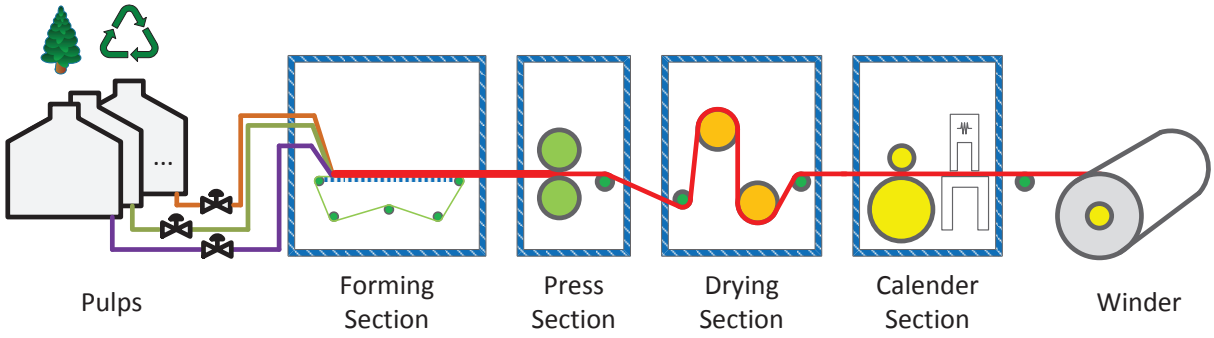


Figure 7.10: The principle of paper production

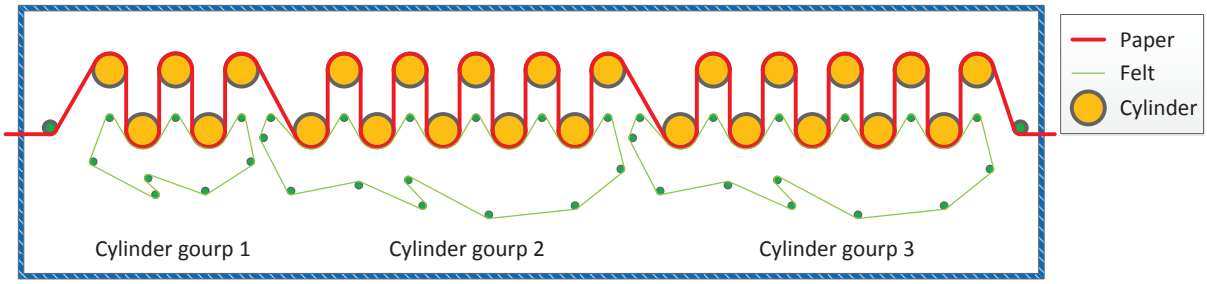


Figure 7.11: The paper drying section for the case studies

7.3 Case studies on the PD benchmark process

7.3.1 Process description

Paper and paper-based products play an important role in our daily life. Industrial paper-making is known to have been traced back to China to the year 105 and today it is a highly competitive and capital-intensive market that is under increasing price pressure. The principle of paper-making is simple. As shown in Figure 7.10, pulps are firstly produced from trees or recycled materials. When they enter the forming section, the water content is around 99% [103]. In the forming section, the pulps are dispensed through a long slice onto the wire, where the width of wet paper can reach 7 meters. When the paper leaves the forming section, around 19% water can be removed. The wet paper sheet then passes through the press section, where around 30% water is squeezed out by large rolls loaded under high pressure. To further remove the water content, the paper sheet is fed to a drying section. Among different drying techniques, steam-heated cylinders are widely used. In modern automatic processes, the number of cylinders can reach 70. From technical viewpoint, the drying section is the most important section for

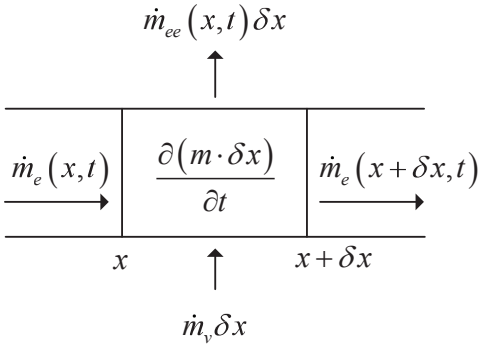


Figure 7.12: The control volume of the paper sheet [8]

the paper quality in a paper machine. By manipulating the temperature of the wet paper sheet, most of the remaining water is removed through evaporation. When the paper sheet leaves the drying section, it contains only about 5% water. Afterwards, the paper sheet passes through the calender section, where the surfaces of the dry paper sheet is smoothed and thus its brightness is improved. It is important to mention that many important quality indicators like moisture and basis weight are measured in the calender section. Finally, the dry paper sheet is rolled by the winders. After trimming both sides, the paper rolls are shipped to customers.

For the case studies in this section, we will focus on the drying section. The moisture is defined as a KPI, which is [8]

$$H = \frac{m_e}{m_p} = \frac{m}{m_p} - 1 \text{ [kg water/kg fibers]}$$

where m_e denotes the mass of water in a paper section, m_p denotes the mass of fibers and m is the total mass of the paper section. The manipulated variable is the paper temperature (or the surface pressure of the wet paper sheet), which is continuously distributed along the machine direction. Thus to model the drying process, the ODE fails. In practice, multiple steam-heated cylinders are used as actuators for the drying process. As shown in Figure 7.11, there are 25 steam-heated cylinders divided into three groups. The first cylinder group is used to warm up the paper sheet where less water is removed. The other two groups remove most of the water. To model the dynamic relationship between the manipulated variable and the moisture, we adopt the first principles PDE model for the paper sheet given in [8, 9], which is also used in our early study [45]. Figure 7.12 shows the mass balance for a paper section of length δx, where x denotes the machine direction. Due to the movement of the paper sheet to the right, water (in the paper) flows into the control volume with a mass flow of $\dot{m}_e(x,t)$ from the left side and flows out from the right side with a mass flow of $\dot{m}_e(x+\delta x,t)$. We assume the mass of fibers is not

changed. Heated by the hot cylinders, the paper temperature increases and therefore the water is removed through evaporation. The mass flow is denoted as $\dot{m}_{ee}(x,t)$. Meanwhile, the vapor flows back to the paper through deposition, whose mass flow is denoted as $\dot{m}_v$, which can be considered as constant. By applying the mass conservation law for the control volume, we have

$$\dot{m}_e(x,t) - \dot{m}_e(x+\delta x,t) - \dot{m}_{ee}(x,t)\delta x + \dot{m}_v\delta x = \frac{\partial(m\ \delta x)}{\partial t}. \tag{7.1}$$

In addition, a description of the relationship between the evaporation rate and water vapor pressure is required, which is given as

$$\dot{m}_{ee}(x,t) = \beta(x)(P(x,t) - P_a) \tag{7.2}$$

where P_a denotes the partial pressure of water in the atmosphere and is given by the temperature and humidity of the air, $\beta(x)$ is the mass transfer coefficient, and $P(x,t)$ denotes the surface pressure of the wet paper sheet and depends on paper temperature. It is further assumed that $\dot{m}_e(x,t) = m_e(x,t)v$ where v represents the constant machine speed. By submitting (7.2), $m_e(x,t) = m_p H(x,t)$ and $m(x,t) = m_p(H(x,t)+1)$ into (7.1) we can get

$$\frac{\partial H(x,t)}{\partial t} = -v\frac{\partial H(x,t)}{\partial x} + \frac{\dot{m}_v}{m_p} - \frac{\beta(x)}{m_p}(P(x,t) - P_a).$$

To determine $\beta(x)$, let us divide the paper sheet in the drying section into three types of zones as

- Zone 1: Unfelted zones which are the contact areas of the paper sheet with the upper cylinders in Figure 7.11,
- Zone 2: Contact with air on both sides zones where the paper sheet is not contacting any cylinder, and
- Zone 3: Felted zones which are the contact areas of the paper sheet with the lower cylinders in Figure 7.11.

Based on it, we have

$$\beta(x) = \begin{cases} \beta_a, & \text{zone 1;} \\ 2\beta_a, & \text{zone 2;} \\ \beta_h, & \text{zone 3} \end{cases}$$

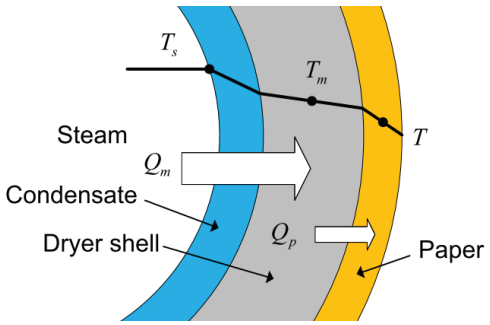


Figure 7.13: A piece of the cross-section of a drying cylinder [103]

where β_a denotes the approximate mass-transfer coefficient between and paper sheet and air, and β_h denotes the approximate mass-transfer coefficient between felt and paper sheet. In addition, we compute $P(x,t)$ from $T(x,t)$ as

$$P(x,t) = 50.104 \times 10^6 e^{-4892.5/T(x,t)}$$

which is a static equation and $T(x,t)$ denotes the paper temperature. For simulating the process in this section, we assume the static relationship holds for the whole dryer section. Thus $P(x,t)$ or $T(x,t)$ can be considered as the manipulated variable. To achieve an actuator model (heat transfer from steam to the paper sheet), we refer to the first principles ODE model given in [103]. The heating principle is shown in Figure 7.13. Hot steam is fed into the cylinder. When it condenses on the inner surface of the cylinder, heat transfers from the steam to the condensate and further to the metal, which is denoted as Q_m. The water which contains less energy is removed by siphons. Meanwhile, the heat transfers further from the metal to the paper, which is denoted as Q_p. The temperature of the condensate, the metal shell and the paper is denoted as T_s, T_m and T, respectively. By applying mass and energy balance laws, the following equations can be derived

$$h_s(p)V\frac{d\rho_s}{dp}\frac{dp}{dt} = q_s h_s(p) - q_w(p)h_w(p) - \alpha_{sc}A_{cyl}(T_s(p) - T_m)$$
$$mC_{p,m}\frac{dT_m}{dt} = \alpha_{sc}A_{cyl}(T_s(p) - T_m) - \alpha_{cp}A_{cyl}\eta(T_m - T)$$

where p denotes the steam pressure, V is the cylinder volume, h_s and h_w are the steam and water enthalpy, ρ_s is the steam density, q_s and q_w denote mass flow rate of the steam into the cylinder and the siphon flow rate, α_{sc} and α_{cp} are the heat transfer coefficients from the steam-condensate to the centre of the cylinder shell and from the centre of the cylinder shell to the centre of the paper, respectively, A_{cyl} is the inner cylinder area, m and $C_{p,m}$ are the mass and specific heat capacity of the shell, and η is the fraction of dryer

Table 7.3: PD: Numerical values of the parameters used in the case studies [8, 103]

Notation	Value	Units	Notation	Value	Units
v	8.33	m/s	l	25	m
$H(0,t)$	50%	–	Dryer roll	25	–
p^0	90	kPa	$T_s(p^0)$	369.84	K
h_s	2670.3	kJ/kg	h_w	405.2	kJ/kg
α_{sc}	1.80	$kW/(m^2K)$	α_{cp}	1.20	$kW/(m^2K)$
η	0.5	–	A_cyl	37.2	m^2
m_p	0.192	kg/m^2	$\dot{m}_v$	0.0005	$kg/(ms)$
β_a	0.00209	–	β_h	0.0011	–

surface covered by the paper web. The equilibrium gives the relation

$$T = T_s(p^0) - \left(\frac{1}{\alpha_{sc}} + \frac{1}{\alpha_{cp}\eta}\right)\frac{1}{A_{cyl}}(q_s^0 h_s(p^0) - q_w^0 h_w(p^0))$$

where q_s^0 is the input and the other parameters can be obtained from process knowledge. Since our focus is on the performance monitoring, the above equation is considered as the actuator model for simplicity. As a result, the mathematical model of the drying paper can be written as

$$\frac{\partial H(x,t)}{\partial t} = -v\frac{\partial H(x,t)}{\partial x} + y(x,t) + \eta(x,t), x \in [0,l] \tag{7.3}$$

where $\eta(x,t)$ represents the process noise, l is the paper length in the drying section, and

$$y(x,t) = \frac{\dot{m}_v}{m_p} - \frac{\beta(x)}{m_p}(P(x,t) - P_a)$$

with the Dirichlet boundary condition $H(0,t) = 50\%$ on the wet end.

By putting the spatial variable in dimensionless form, $x' = x/l$, we obtain

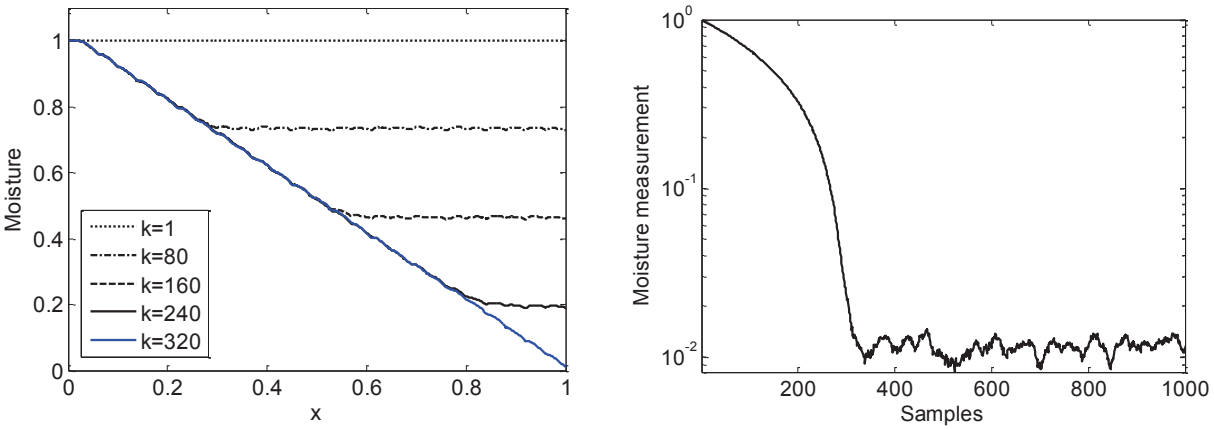
$$\frac{\partial H(x',t)}{\partial t} = -\frac{v}{l}\frac{\partial H(x',t)}{\partial x'} + y(x',t) + \eta(x',t), x' \in [0,1]. \tag{7.4}$$

For notational simplicity, we will neglect the prime symbol in the following.

7.3.2 Detection of two typical fault episodes

Based on the process and actuator models described above, we will apply Algorithm 5.4 (for the common industrial configuration) proposed in Chapter 5 to detect two typical fault episodes of the drying process. To simulate the drying process, we use the parameters given in Table 7.3[2]

[2]The property of saturated steam is obtained from $http://www.efunda.com/Materials/water/steam-table\_sat.cfm$.



(a) Evolution of moisture profile (b) Moisture measurement

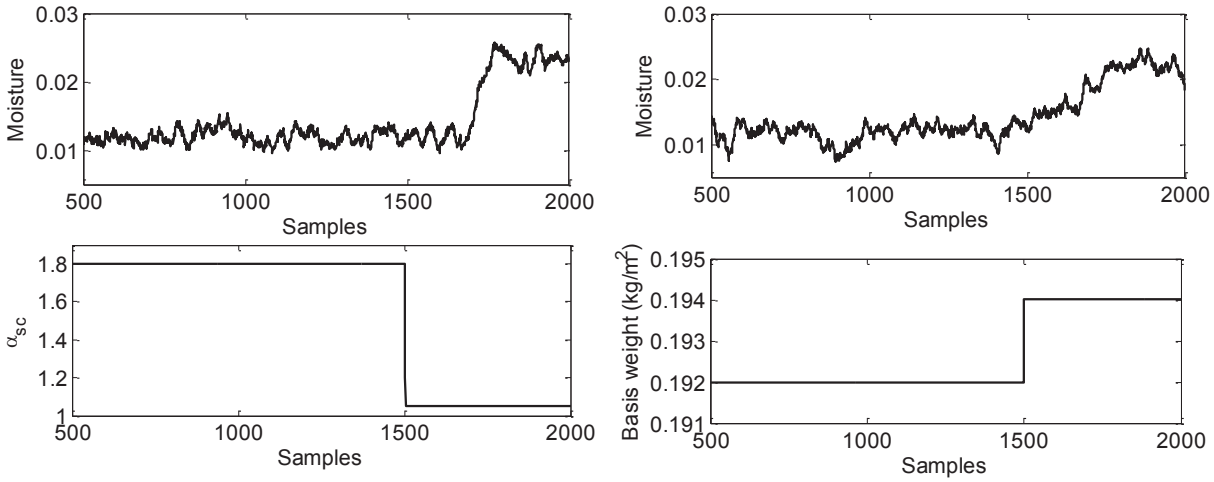
Figure 7.14: Illustration of the drying process

and 101 polynomial basis functions. The sampling time is chosen to be 0.01 . The moisture is measured at the dry end. Figures 7.14-a-b show the simulation results in the fault-free case. From the left plot we can see the evolution of the moisture profile of the paper in the drying section. Initially no heat is transferred from the steam to the paper sheet, thus the moisture profile is constant. Then the water evaporates from the paper and at the 320th sample, the moisture profile is in the steady state. Figure 7.14-b shows the moisture measurement. After about 300 samples, the dry paper with an average moisture value of 1.17% is produced (Gaussian process and measurement noise is included).

To show the monitoring performance of the data-driven method proposed in Chapter 5, we simulated two typical fault episodes that happen during the drying process, *i.e.*

- *Inner cylinder wall fouling:* Similar to the *heat exchanger fouling* episode discussed previously, the deposited solid on the inner wall will decrease the heat transfer efficiency. To this end, the heat transfer coefficient from the condensate to the cylinder shell is reduced by 0.75 for the 7th cylinder from the 1500th sample. As shown in Figure 7.15-a, this fault influences the moisture measurement after about 200 samples (around 2 s). This kind of fault is quite common for the drying section. In industrial applications, the steam consumption will be significantly increased. As a result, detection of this kind of fault is very important to keep a low operation cost.

- *Increase of the basis weight:* Due to reduction of the machine speed and low performance of the thickness control, the basis weight of the paper sheet could increase. To simulate this kind of fault, we increase the basis weight by 0.002 kg/m^2 from the



(a) Fouling of the inner surface of the 7th drying cylinder

(b) Increase of dry basis weight

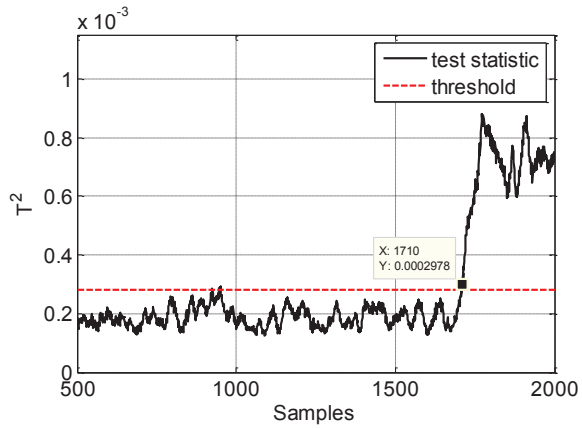
Figure 7.15: PD: Description of the fault episodes

1500th sample. As shown in Figure 7.15-b, the moisture measurement is impacted as well. This kind of fault not only lowers the paper quality, but also consumes more raw material.

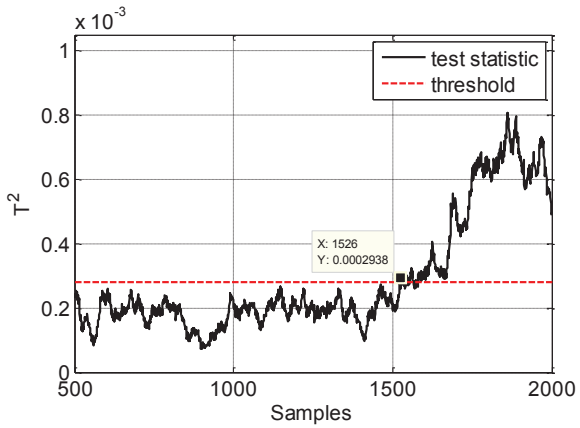
To apply the proposed method, we first run the process without any fault to collect the "snapshot" data for the manipulated variable and the KPI measurement. Then a subspace with 25 basis functions is identified which contains almost 100% of the variability of the "snapshot" data. For identifying the kernel representation, we choose $s = 5$ and $s_p = 7$. Figures 7.16-a-b show the monitoring results where the significance level for the threshold is 0.01. We can see that both faults have been detected. The detection delay of the first fault is 210 samples (2.1 s). This is because the fault happens for the 7th cylinder which is far away from the KPI sensor.

7.4 Concluding remarks

In this chapter, we have used three benchmark processes to demonstrate the performance of the proposed KPI monitoring approaches. The multivariate statistics based methods proposed in Chapters 3 and 6 are applied to the TE process for both fault detection and diagnosis. The signals generated by the TE process are steady and therefore suitable for the static methods which are based on the assumption that the mean values of process variables and KPIs are constant. In addition, the CSTH process, which is dynamic and has varying mean values (depending on the users' consumption), is used to show the



(a) Monitoring result for the fouling of the inner surface of the 7th drying cylinder



(b) Monitoring result for the increase of the dry basis weight

Figure 7.16: PD: KPI monitoring results

effectiveness of the dynamic method given in Chapter 4. Finally, the PD process, whose dynamical description requires a PDE, is used to test the data-driven method proposed in 5. The case study results show that all tested data-driven methods can be used in industrial automation processes.

8 Conclusions and further work

In this thesis, new data-driven KPI monitoring and diagnosis techniques are developed for complex automation processes. In Chapter 1, the background, motivation and the state of the art of the techniques are given. With the increasing global competition, there appears an urgent industrial requirement for KPI monitoring and diagnosis in automation processes. Due to the scale and complexity of automation processes, application of model- or knowledge-based techniques is becoming too expensive or even impossible. As a result, the main objective of this thesis is to develop data-driven KPI monitoring and diagnosis techniques.

The basis of this thesis is given in Chapter 2. Following the mathematical descriptions of the static, lumped-parameter and distributed-parameter processes, basic fault detection techniques are discussed. Among them are statistical approaches including the GLR-, PCA- and PLS-based methods for the static processes, model-based FDF, DO and PS for LPPs, and the eigen-decomposition based method for DPPs. The results achieved in the following chapters are based on them but have improved performance.

Chapter 3 focuses on the KPI monitoring techniques for the static processes. Based on the analysis of the traditional PLS-based method which is not optimal for the KPI monitoring, we first propose a modified approach. This approach is still based on the PLS algorithm, but the monitoring performance is improved. It is important to mention that only two test statistics are involved for monitoring the KPI-related and -unrelated subspaces, respectively. To further reduce the computation cost and engineering effort, an alternative decomposition-based algorithm is given. This algorithm achieves the benefit of the modified approach with much less computation costs. As a result, the alternative algorithm is quite suitable for large-scale automation processes.

Aiming at monitoring dynamic processes where the number of involved variables is small, Chapter 4 presents a data-driven dynamic monitoring approach using the subspace identification method. Compared with the standard approaches, this method requires much less engineering effort. The monitoring system is directly identified from the process I/O data without identifying a system model.

In Chapter 5, novel model-based and data-driven KPI monitoring techniques are developed for DPPs. These methods are suitable for processes where the spatial dynamics

is dominant, *e.g.* the paper drying process and the hot strip mill. Different from the existing techniques, the proposed monitoring approaches are developed for random processes. They are more realistic and can directly be used in the real processes. Moreover, the new methods are not based on the eigen-decomposition of the system operators and thus are easier to understand. If an actual process model is easy to derive, it is advised to use the model-based version which can achieve better performance. Otherwise, the data-driven version can be used, which only requires the historical process and KPI data. It is important to mention that the performance of the data-driven method is depending on the quality of the data. The more informative the data are, the better the monitor system performs.

After performance degradation is detected, it is urgent to find out its root cause and do the corrective actions. Chapter 6 proposes a novel data-driven performance diagnosis method. Different from the existing approaches, this method is able to diagnose the multiplicative fault which is more complex and costly (for diagnosis). It is based on process data and requires no process knowledge. It aims at assisting the process engineers by narrowing down the investigation scope.

Finally, the algorithms developed in Chapters 3-6 are tested on three realistic industrial benchmark processes in Chapter 7. The test results show that the proposed methods are quite suitable for practical applications.

The results achieved in this thesis are based on the linear system descriptions. They are efficient if the real process is working around the operating point. However, there are nonlinear systems which are working in a large operation range. Extension of the proposed methods to these processes are of practical importance and requires more research attention. In addition, integration of the monitoring and diagnosis results into the control system is a great challenge and can be addressed in future work.

A Proof of Theorem 2.1

Proof. For $i = 1$, it holds for

$$\mathbf{Y}_j = \mathbf{Y}_{j-i} - \mathbf{p}_{j-i}\mathbf{t}_{j-i}^T = \mathbf{Y}_{j-i}\left(\mathbf{I} - \frac{\mathbf{t}_{j-i}\mathbf{t}_{j-i}^T}{||\mathbf{t}_{j-i}||^2}\right), j > i.$$

Based on it, we can derive

$$\mathbf{Y}_j\mathbf{t}_{j-i} = \mathbf{Y}_{j-i}\mathbf{t}_{j-i} - \mathbf{Y}_{j-i}\mathbf{t}_{j-i}\frac{\mathbf{t}_{j-i}^T\mathbf{t}_{j-i}}{||\mathbf{t}_{j-i}||^2} = \mathbf{0},$$

thus

$$\mathbf{t}_{j-i}^T\mathbf{t}_j = \mathbf{t}_{j-i}^T\mathbf{Y}_j^T\mathbf{w}_j^* = 0, \Rightarrow \mathbf{t}_j^T\mathbf{t}_{j-1} = \mathbf{t}_{j-1}^T\mathbf{t}_j = 0.$$

Assume that for $i = k$ $(k > 1)$, the following statements hold:

$$\mathbf{t}_j^T\mathbf{t}_{j-k} = 0,\ \mathbf{t}_j^T\mathbf{t}_{j-(k-1)} = 0, \cdots,\ \mathbf{t}_j^T\mathbf{t}_{j-2} = 0, j > k.$$

Then for $i = k + 1$:

$$\mathbf{Y}_j = \mathbf{Y}_{j-1}\left(\mathbf{I} - \frac{\mathbf{t}_{j-1}\mathbf{t}_{j-1}^T}{||\mathbf{t}_{j-1}||^2}\right) = \cdots = \mathbf{Y}_{j-k-1}\prod_{l=k+1}^{1}\left(\mathbf{I} - \frac{\mathbf{t}_{j-l}\mathbf{t}_{j-l}^T}{||\mathbf{t}_{j-l}||^2}\right)$$

we have

$$\begin{aligned}\mathbf{Y}_j\mathbf{t}_{j-k-1} &= \mathbf{Y}_{j-k-1}\prod_{l=k+1}^{1}\left(\mathbf{I} - \frac{\mathbf{t}_{j-l}\mathbf{t}_{j-l}^T}{||\mathbf{t}_{j-l}||^2}\right)\mathbf{t}_{j-k-1} \\ &= \mathbf{Y}_{j-k-1}\left(\mathbf{I} - \frac{\mathbf{t}_{j-k-1}\mathbf{t}_{j-k-1}^T}{||\mathbf{t}_{j-k-1}||^2}\right)\left(\mathbf{I} - \frac{\mathbf{t}_{j-k}\mathbf{t}_{j-k}^T}{||\mathbf{t}_{j-k}||^2}\right)\cdots\left(\mathbf{I} - \frac{\mathbf{t}_{j-2}\mathbf{t}_{j-2}^T}{||\mathbf{t}_{j-2}||^2}\right)\left(\mathbf{I} - \frac{\mathbf{t}_{j-1}\mathbf{t}_{j-1}^T}{||\mathbf{t}_{j-1}||^2}\right)\mathbf{t}_{j-k-1}\end{aligned}$$

Since from $i = k$ we have assumed

$$\mathbf{t}_{j-1}^T\mathbf{t}_{j-1-k} = 0,\ \mathbf{t}_{j-2}^T\mathbf{t}_{j-2-(k-1)} = 0,\ \cdots,\ \mathbf{t}_{j-k+1}^T\mathbf{t}_{j-k+1-2} = 0$$

and from $i = 1$ we have obtained $\mathbf{t}_{j-k}^T \mathbf{t}_{j-k-1} = 0$, thus

$$\begin{aligned}
\mathbf{Y}_j \mathbf{t}_{j-k-1} &= \mathbf{Y}_{j-k-1} \left(\mathbf{I} - \frac{\mathbf{t}_{j-k-1} \mathbf{t}_{j-k-1}^T}{||\mathbf{t}_{j-k-1}||^2} \right) \mathbf{t}_{j-k-1} \\
&= \mathbf{Y}_{j-k-1} \mathbf{t}_{j-k-1} - \mathbf{Y}_{j-k-1} \mathbf{t}_{j-k-1} \frac{\mathbf{t}_{j-k-1}^T \mathbf{t}_{j-k-1}}{||\mathbf{t}_{j-k-1}||^2} = 0 \\
&\Rightarrow \mathbf{t}_{j-k-1}^T \mathbf{t}_j = \mathbf{t}_{j-k-1}^T \mathbf{Y}_j^T w_j^* = 0 = \mathbf{t}_j^T \mathbf{t}_{j-k-1}.
\end{aligned}$$

which indicates that the assumptions for $i = k$ also holds for $i = k + 1$,i.e.

$$\mathbf{t}_j^T \mathbf{t}_{j-k-1} = 0,\ \mathbf{t}_j^T \mathbf{t}_{j-k} = 0,\ \mathbf{t}_j^T \mathbf{t}_{j-(k-1)} = 0, \cdots,\ \mathbf{t}_j^T \mathbf{t}_{j-2} = 0, j > k + 1$$

Thus for any $i = 1, 2, \cdots, k, k + 1, \cdots, \infty$, we have $\mathbf{t}_j^T \mathbf{t}_{j-i} = 0$. □

B Theorems for Chapter 5

Theorem B.1. *Given a full column-rank matrix* $\mathbf{P} \in \mathcal{R}^{n\times\gamma}$ *and an arbitrary vector* $\mathbf{q} \in \mathcal{R}^n$*, finding a* $\hat{\mathbf{q}}$ *satisfying (5.7) is equivalent to find a* $\hat{\mathbf{q}} \in \hat{\mathcal{P}} := span\{\mathbf{p}_1, \cdots, \mathbf{p}_\gamma\}$ *such that*

$$(\mathbf{q} - \hat{\mathbf{q}})^T \mathbf{p}_i = 0, \text{ for } i = 1, \cdots, \gamma.$$

Proof. Suppose that $r = \hat{\mathbf{q}} + t\boldsymbol{\Delta}\mathbf{q}$ where $t \in \mathcal{R}$ and $\boldsymbol{\Delta}\mathbf{q} \in \hat{\mathcal{P}}$. The meaning of (5.7) is that $\hat{\mathbf{q}} \in \hat{\mathcal{P}}$ is closest to $\mathbf{q}$ if and only if

$$\begin{aligned}(\mathbf{q} - \hat{\mathbf{q}})^T(\mathbf{q} - \hat{\mathbf{q}}) &\leq (\mathbf{q} - (\hat{\mathbf{q}} + t\boldsymbol{\Delta}\mathbf{q}))^T(\mathbf{q} - (\hat{\mathbf{q}} + t\boldsymbol{\Delta}\mathbf{q})) \\ &= (\mathbf{q} - \hat{\mathbf{q}})^T(\mathbf{q} - \hat{\mathbf{q}}) - 2t(\mathbf{q} - \hat{\mathbf{q}})^T\boldsymbol{\Delta}\mathbf{q} + t^2\boldsymbol{\Delta}\mathbf{q}^T\boldsymbol{\Delta}\mathbf{q}, \ \forall\boldsymbol{\Delta}\mathbf{q} \in \hat{\mathcal{P}} \ \& \ \forall t \in \mathcal{R}\end{aligned}$$

which is equivalent to $-2t(\mathbf{q} - \hat{\mathbf{q}})^T\boldsymbol{\Delta}\mathbf{q} + t^2\boldsymbol{\Delta}\mathbf{q}^T\boldsymbol{\Delta}\mathbf{q} \geq 0, \ \forall\boldsymbol{\Delta}\mathbf{q} \in \hat{\mathcal{P}} \ \& \ \forall t \in \mathcal{R}$.

The above equation has two variables. If we consider $\boldsymbol{\Delta}\mathbf{q}$ as any fixed vector, then the inequality only holds when

$$(\mathbf{q} - \hat{\mathbf{q}})^T\boldsymbol{\Delta}\mathbf{q} = 0, \forall\boldsymbol{\Delta}\mathbf{q} \in \hat{\mathcal{P}}. \tag{B.1}$$

The equivalence of (5.7) and (B.1) can be easily revealed as:

- sufficiency: If (B.1) holds, then (5.7) holds as well since $\mathbf{p}_i, i = 1, \cdots, \gamma \in \hat{\mathcal{P}}$
- necessity: If (5.7) holds, then, since $\{\mathbf{p}_1, \cdots, \mathbf{p}_\gamma\}$ is a basis for $\hat{\mathcal{P}}$, any $\boldsymbol{\Delta}\mathbf{q} \in \hat{\mathcal{P}}$ can be represented as $\boldsymbol{\Delta}\mathbf{q} = \sum_{i=1}^{\gamma} z_i^*\mathbf{p}_i, z_i^* \in \mathcal{R}, i = 1, \cdots, \gamma$. (B.1) can be written as

$$\begin{aligned}(\mathbf{q} - \hat{\mathbf{q}})^T\boldsymbol{\Delta}\mathbf{q} &= (\mathbf{q} - \hat{\mathbf{q}})^T \sum_{i=1}^{\gamma} z_i^*\mathbf{p}_i \\ &= \sum_{i=1}^{\gamma} z_i^* \left((\mathbf{q} - \hat{\mathbf{q}})^T\mathbf{p}_i\right) \\ &= \sum_{i=1}^{\gamma} z_i^* 0 = 0, \forall\boldsymbol{\Delta}\mathbf{q} \in \hat{\mathcal{P}}.\end{aligned}$$

□

Theorem B.2. *Given an arbitrary function $q(x) \in \mathcal{H}$ and a subspace $\mathcal{P}(x)$ spanned by $\mathbf{p}(x) = [p_1(x), \cdots, p_\gamma(x)]^T$, find a $\hat{q}(x)$ in $\mathcal{P}(x)$ satisfying*

$$\int_\alpha^\beta \left(q(x) - \hat{q}(x)\right)^2 dx = \min_{r(x) \in span\{\mathbf{p}(x)\}} \int_\alpha^\beta \left(q(x) - r(x)\right)^2 dx \tag{B.2}$$

is equivalent to find a $\hat{q}(x)$ in $\mathcal{P}(x)$ such that

$$\int_\alpha^\beta \left(q(x) - \hat{q}(x)\right) p_i(x) dx = 0 \text{ for } i = 1, \cdots, \gamma. \tag{B.3}$$

Proof. The basic idea of proof is similar to the proof for Theorem B.1. Suppose that $r(x) = \hat{q}(x) + t\Delta q(x)$ where $t \in \mathcal{R}$ and $\Delta q(x) \in \mathcal{P}(x)$. From (B.2) we have

$$\int_\alpha^\beta (q(x) - \hat{q}(x))^2 dx \leq \int_\alpha^\beta (q(x) - (\hat{q}(x) + t\Delta q(x)))^2 dx$$

$$= \int_\alpha^\beta \left((q(x) - \hat{q}(x))^2 - 2t(q(x) - \hat{q}(x))\Delta q(x) + t^2 \Delta q(x)^2\right) dx, \ \forall \Delta q(x) \in \mathcal{P}(x) \ \& \ \forall t \in \mathcal{R}$$

which is equivalent to

$$-2t \int_\alpha^\beta (q(x) - \hat{q}(x))\Delta q dx + t^2 \int_\alpha^\beta \Delta q(x)^2 dx \geq 0, \ \forall \Delta q(x) \in \mathcal{P}(x) \ \& \ \forall t \in \mathcal{R}.$$

If we consider $\Delta q(x)$ as any fixed function, then the above equation is in the quadratic form and the inequality only holds when

$$(q(x) - \hat{q}(x)\Delta q(x) = 0, \forall \Delta q(x) \in \mathcal{P}(x). \tag{B.4}$$

The equivalence of (B.3) and (B.4) can be easily revealed as:

- sufficiency: If (B.4) holds, then (B.3) holds as well since $p_i(x), i = 1, \cdots, \gamma \in \mathcal{P}(x)$
- necessity: If (B.3) holds, then, since $\{p_1(x), \cdots, p_\gamma(x)\}$ is a basis for $\mathcal{P}(x)$, any $\Delta q(x) \in \mathcal{P}(x)$ can be represented as $\Delta q(x) = \sum_{i=1}^\gamma z_i^* p_i(x), z_i^* \in \mathcal{R}, i = 1, \cdots, \gamma$. (B.4) can be written as

$$\begin{aligned}
\int_\alpha^\beta (q(x) - \hat{q}(x))\Delta q(x) dx &= \int_\alpha^\beta (q(x) - \hat{q}(x)) \sum_{i=1}^\gamma z_i^* p_i(x) dx \\
&= \sum_{i=1}^\gamma z_i^* \int_\alpha^\beta (q(x) - \hat{q}(x)) p_i(x) dx \\
&= 0, \ \forall \Delta q(x) \in \mathcal{P}(x).
\end{aligned}$$

□

Theorem B.3. *Given are an arbitrary vector* $\mathbf{z}(x,t) \in \mathcal{H}^n$ $(z_i(x,t) \in \mathcal{V}(x), i = 1, \cdots, n)$ *and a known finite dimensional subspace* $\hat{\mathcal{V}}(x) \in span\{v_1(x), \cdots, v_\gamma(x)\} \subset \mathcal{V}(x)$. *The best estimate of* $\mathbf{z}(x,t)$ *from* $\hat{\mathcal{V}}(x)$ *in the sense that*

$$\mathrm{tr}\left(||\mathcal{L}(\mathbf{z}(x,t)) - \mathcal{L}(\hat{\mathbf{z}}(x,t))||\right) = \min_{\mathbf{r}_i(x,t)\in\hat{\mathcal{V}}, i=1,\cdots,n} \mathrm{tr}\left(||\mathcal{L}(\mathbf{z}(x,t)) - \mathcal{L}(\mathbf{r}(x,t))||\right)$$

is equivalent to the solution of

$$\left(\mathcal{L}(\mathbf{z}(x,t) - \hat{\mathbf{z}}(x,t)), v_i(x)\right), \forall\ i = 1, \cdots, \gamma. \tag{B.5}$$

Proof. The proof is similar to the proof for Theorem B.2. □

Bibliography

[1] A. Ajami and M. Daneshvar. Data-driven Approach for Fault Detection and Diagnosis of Turbine in Thermal Power Plant Using Independent Component Analysis (ICA). *International Journal of Electrical Power & Energy Systems*, 43(1):728–735, 2012.

[2] C.F. Alcala and S.J. Qin. Reconstruction-based Contribution for Process Monitoring. *Automatica*, 45(7):1593–1600, 2009.

[3] A. Armaou and M.A. Demetriou. Robust Detection and Accommodation of Incipient Component and Actuator Faults in Nonlinear Distributed Processes. *AIChE Journal*, 54(10):2651–2662, 2008.

[4] A. Baniamerian and K. Khorasani. Fault Detection and Isolation of Dissipative Parabolic PDEs: Finite-dimensional Geometric Approach. In *Proceedings of the 2012 American Control Conference (ACC)*, pages 5894–5899, Montréal, Canada, 2012.

[5] M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, 1993.

[6] R.V. Beard. *Failure Accommodation in Linear Systems through Self-reorganization*. PhD Dissertation, Massachusetts Institute of Technology, USA, 1971.

[7] M. El Hachemi Benbouzid. A Review of Induction Motors Signature Analysis as A Medium for Faults Detection. *IEEE Transactions on Industrial Electronics*, 47(5): 984–993, 2000.

[8] M. Berrada, S. Tarasiewicz, and M.J. Richard. A Computer Model for the Drying Paper in The Paper Production Industry. In *Proceedings of Modeling and Simulation Conference*, pages 2175–2183, Pittsburgh, USA, 1992.

[9] M. Berrada, S. Tarasiewicz, M.E. Elkadiri, and P.H. Radziszewski. A State Model for the Drying Paper in the Paper Production Industry. *IEEE Transactions on Industrial Electronics*, 44(4):579–586, 1997.

[10] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-tolerant Control*. Springer Verlag, 2006.

[11] G.E.P. Box. Some Theorems on Quadratic Forms Applied in The Study of Analysis of Variance Problems, I. Effect of Inequality of Variance in the One-Way Classification. *The Annals of Mathematical Statistics*, 25(2):290–302, 1954.

[12] C.C. Chang and C.C. Yu. On-line Fault Diagnosis Using The Signed Directed Graph. *Industrial & Engineering Chemistry Research*, 29(7):1290–1299, 1990.

[13] P.S. Chavez and A.Y. Kwarteng. Extracting Spectral Contrast in Landsat Thematic Mapper Image Data Using Selective Principal Component Analysis. *Photogrammetric Engineering & Remote Sensing*, 55(3):339–348, 1989.

[14] J. Chen and R.J. Patton. *Robust Model-based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

[15] J. Chen, R.J. Patton, and H.Y. Zhang. Design of Unknown Input Observers and Robust Fault Detection Filters. *International Journal of Control*, 63(1):85–105, 1996.

[16] R.H. Chen, D.L. Mingori, and J.L. Speyer. Optimal Stochastic Fault Detection Filter. *Automatica*, 39:377–390, 2003.

[17] L.H. Chiang, E.L. Russell, and R.D. Braatz. Fault Diagnosis in Chemical Processes Using Fisher Discriminant Analysis, Discriminant Partial Least Squares, and Principal Component Analysis. *Chemometrics & Intelligent Laboratory Systems*, 50(2): 243–252, 2000.

[18] L.H. Chiang, E. Russell, and R.D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer Verlag, London, 2001.

[19] P.D. Christofides. *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport Reaction Processes*. Birkhäuser, 2001.

[20] R.F. Curtain and H.J. Zwart. *An Introduction to Infinite-dimensional Linear Systems Theory*. Springer Verlag, 1995.

[21] M.A. Demetriou and A. Armaou. Robust Detection and Accommodation of Incipient Component Faults in Nonlinear Distributed Processes. In *Proceedings of the 2007 American Control Conference (ACC)*, pages 2054–2059, New York, USA, 2007.

[22] M.A. Demetriou and K. Ito. Online Fault Detection and Diagnosis for A Class of Positive Real Infinite Dimensional Systems. In *Proceedings of the 41st IEEE Conference on Decision and Control (CDC)*, pages 4359–4364, Las Vegas, USA, 2002.

[23] S.X. Ding. Data-driven Design of Model-based Fault Diagnosis Systems. In *Proceedings of the 8th IFAC International Symposium on Advanced Control of Chemical Processes*, pages 1323–1328, Mexico City, Mexico, 2012.

[24] S.X. Ding. *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms and Tools*. Springer Verlag, second edition, 2013.

[25] S.X. Ding, P. Zhang, E. L. Ding, P. M. Frank, and M. Sader. Multiobjective Design of Fault Detection Filters. In *Proceedings of the 2003 European Control Conference (ECC)*, Cambridge, UK, 2003.

[26] S.X. Ding, P. Zhang, A. Naik, E.L. Ding, and B. Huang. Subspace Method Aided Data-driven Design of Fault Detection and Isolation Systems. *Journal of Process Control*, 19(9):1496–1510, 2009.

[27] S.X. Ding, P. Zhang, E.L. Ding, S. Yin, A. Naik, P.C. Deng, and W.H. Gui. On The Application of PCA Technique to Fault Diagnosis. *Tsinghua Science & Technology*, 15(2):138–144, 2010.

[28] S.X. Ding, S. Yin, K.X. Peng, H.Y. Hao, and B. Shen. A Novel Scheme for Key Performance Indicator Prediction and Diagnosis With Application to An Industrial Hot Strip Mill. *IEEE Transactions on Industrial Informatics*, 9(4):2239–2247, 2013.

[29] X. Ding, L. Guo, and T. Jeinsch. A Characterization of Parity Space and Its Application to Robust Fault Detection. *IEEE Transactions on Automatic Control*, 44(2):337–343, 1999.

[30] J.J. Downs and E.F. Vogel. A Plant-wide Industrial Process Control Problem. *Computers & Chemical Engineering*, 17(3):245–255, 1993.

[31] B.M. Ebrahimi, J. Faiz, and B. Akin. Pattern Recognition for Broken Bars Fault Diagnosis in Induction Motors under Various Supply Conditions. *European Transactions on Electrical Power*, 22(8):1176–1190, 2012.

[32] N.H. El-Farra. Integrated Fault Detection and Fault-tolerant Control Architectures for Distributed Processes. *Industrial & Engineering Chemistry Research*, 45(25): 8338–8351, 2006.

[33] P.M. Frank. Fault Diagnosis in Dynamic Systems using Analytical and Knowledge-based Redundancy: A Survey and Some New Results. *Automatica*, 26(3):459–474, 1990.

[34] P.M. Frank. Online Fault-Detection in Unknown Nonlinear-Systems Using Diagnostic Observers - A Survey. *International Journal of System Science*, 25(12):2129–2154, 1994.

[35] P.M. Frank. Analytical and Qualitative Model-based Fault Diagnosis: A Survey and Some New Results. *European Journal of Control*, 2(1):6–28, 1996.

[36] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.

[37] J. Gertler and J. Cao. PCA-based Fault Diagnosis in The Presence of Control and Dynamics. *AIChE Journal*, 50(2):388–402, 2004.

[38] J. Gertler, W.H. Li, Y.B. Huang, and T. McAvoy. Isolation Enhanced Principal Component Analysis. *AIChE Journal*, 45(2):323–334, 1999.

[39] M.S. Gockenbach. *Partial Differential Equations: Analytical and Numerical Methods*. SIAM, 2002.

[40] Fredrik Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.

[41] A. Haghani. *Data-driven Design of Fault Diagnosis Systems for Nonlinear Multimode Processes*. PhD Dissertation, University of Duisburg-Essen, Germany, 2013.

[42] A. Haghani, S.X. Ding, H.Y. Hao, S. Yin, and T. Jeinsch. An Approach for Multimode Dynamic Process Monitoring using Bayesian Inference. In *Proceedings the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)*, pages 1011–1016, Mexico City, Mexico, 2012.

[43] H.Y. Hao. *Study on Independent Component Analysis (ICA) based Process Monitoring*. Master Thesis, University of Duisburg-Essen, Germany, 2010.

[44] H.Y. Hao, S.X. Ding, A. Haghani, S. Yin, and T. Jeinsch. Influence of Additive and Multiplicative Faults on Process Output Variances. In *Proceedings of the 1st PAPYRUS Workshop on Fault Diagnosis and Fault Tolerant Control in Large Scale Process Monitoring*, Corsica, France, 2011.

[45] H.Y. Hao, S.X. Ding, A. Haghani, and S. Yin. An Observer-based Fault Detection Scheme for Distributed Parameter Systems of Hyperbolic Type and Its Application in Paper Production Process. In *Proceedings the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)*, pages 1047–1052, Mexico City, Mexico, 2012.

[46] H.Y. Hao, K. Zhang, S.X. Ding, Z.W. Chen, Y.G. Lei, and Z.K. Hu. A KPI-related Multiplicative Fault Diagnosis Scheme for Industrial Processes. In *Proceedings of the 10th IEEE International Conference on Control and Automation (ICCA)*, pages 1460–1465, Hangzhou, China, 2013.

[47] H.Y. Hao, K. Zhang, S.X. Ding, Z.W. Chen, and Y.G. Lei. A Data-driven Multiplicative Fault Diagnosis Approach for Automation Processes. *Accepted by ISA Transactions*, 2014.

[48] H. Hotelling. Relations Between Two Sets of Variables. *Biometrika*, 28(3-4):321–377, 1936.

[49] Q. Hu, Z.J. He, Z.S. Zhang, and Y.Y. Zi. Fault Diagnosis of Rotating Machinery Based on Improved Wavelet Package Transform and SVMs Ensemble. *Mechanical Systems & Signal Processing*, 21(2):688–705, 2007.

[50] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[51] Germany TRADE & INVEST. Germany's Chemical and Related Process Industry 2011: A Profile of Selected Investment Sites. 2011.

[52] R. Isermann. Process Fault Detection Based on Modeling and Estimation Methods: A Survey. *Automatica*, 20(4):387–404, 1984.

[53] R. Isermann. *Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Verlag, 2006.

[54] J.W. Jackson and G.S. Mudholkar. Control Procedures for Residuals Associated with Principal Component Analysis. *Technometrics*, 21(3):341–349, 1979.

[55] S.L. Jämsä-Jounela, M. Vermasvuori, P. Endén, and S. Haavisto. A Process Monitoring System Based on the Kohonen Self-organizing Maps. *Control Engineering Practice*, 11(1):83–92, 2003.

[56] A.K.S. Jardine, D.M. Lin, and D. Banjevic. A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance. *Mechanical Systems & Signal Processing*, 20(7):1483–1510, 2006.

[57] C.D. Johnson. *Process Control Instrumentation Technology*. Prentice-Hall, sixth edition, 2000.

[58] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, second edition edition, 1982.

[59] H.L. Jones. *Failure Detection in Linear Systems*. PhD Dissertation, Massachusetts Institute of Technology, USA, 1973.

[60] B.C. Juricek, D.E. Seborg, and W.E. Larimore. Fault Detection Using Canonical Variate Analysis. *Industrial & Engineering Chemistry Research*, 43(2):458–474, 2004.

[61] B. Köppen-Seliger. *Fehlerdiagnose mit künstlichen neuronalen Netzen*. VDI Verlag, 1997.

[62] M.A. Kramer and B.L. Palowitch. A Rule-based Approach to Fault Diagnosis Using The Signed Directed Graph. *AIChE Journal*, 33(7):1067–1078, 1987.

[63] J.V. Kresta, J.F. Macgregor, and T.E. Marlin. Multivariate Statistical Monitoring of Process Operating Performance. *The Canadian Journal of Chemical Engineering*, 69(1):35–47, 1991.

[64] U. Kruger and G. Dimitriadis. Diagnosis of Process Faults in Chemical Systems Using A Local Partial Least Squares Approach. *AIChE Journal*, 54(10):2581–2596, 2008.

[65] U. Kruger, S. Kumar, and T. Littler. Improved Principal Component Monitoring Using the Local Approach. *Automatica*, 43(9):1532–1542, 2007.

[66] W.F. Ku, R.H. Storer, and C. Georgakis. Disturbance Detection and Isolation by Dynamic Principal Component Analysis. *Chemometrics & Intelligent Laboratory Systems*, 30(1):179–196, 1995.

[67] J.H. Lee and A.W. Dorsey. Monitoring of Batch Processes Through State-space Models. *AIChE Journal*, 50(6):1198–1210, 2004.

[68] J.M. Lee, C. Yoo, and I.B. Lee. Statistical Process Monitoring with Independent Component Analysis. *Journal of Process Control*, 14(5):467–485, 2004.

[69] J.M. Lee, S.J. Qin, and I.B. Lee. Fault Detection and Diagnosis Based on Modified Independent Component Analysis. *AIChE Journal*, 52(10):3501–3514, 2006.

[70] W.S. Lee, D.L. Grosh, F.A. Tillman, and C.H. Lie. Fault Tree Analysis, Methods, and Applications: A Review. *IEEE Transactions on Reliability*, R-34(3):194–203, 1985.

[71] Y.G. Lei, D. Han, J. Lin, and Z.J. He. Planetary Gearbox Fault Diagnosis Using an Adaptive Stochastic Resonance Method. *Mechanical Systems & Signal Processing*, 38(1):113–124, 2013.

[72] Y.G. Lei, J. Lin, Z.J. He, and M.J. Zuo. A Review on Empirical Mode Decomposition in Fault Diagnosis of Rotating Machinery. *Mechanical Systems & Signal Processing*, 35(1-2):108–126, 2013.

[73] G. Li, S.J. Qin, and D.H. Zhou. Geometric Properties of Partial Least Squares for Process Monitoring. *Automatica*, 46(1):204–210, 2010.

[74] W.H. Li, Z.G. Han, and S.L. Shah. Subspace Identification for FDI in Systems with Non-uniformly Sampled Multirate Data. *Automatica*, 42(4):619–627, 2006.

[75] S.H. Liao. Expert System Methodologies and Applications: A Decade Review from 1995 to 2004. *Expert Systems with Applications*, 28(1):93–103, 2005.

[76] N. Lü, Z.H. Xiong, X. Wang, and C.R. Ren. Integrated Framework of Probabilistic Signed Digraph Based Fault Diagnosis Approach to A Gas Fractionation Unit. *Industrial & Engineering Chemistry Research*, 50(17):10062–10073, 2011.

[77] H.V. Ly and H.T. Tran. Proper Orthogonal Decomposition for Flow Calculations and Optimal Control in A Horizontal CVD Reactor. *Quarterly of Applied Mathematics*, 60(4):631–656, 1998.

[78] J.F. MacGregor, C. Jaeckle, C. Kiparissides, and M. Koutoudi. Process Monitoring and Diagnosis by Multiblock PLS Methods. *AIChE Journal*, 40(5):826–838, 1994.

[79] J.F. Magni and P. Mouyon. On Residual Generation by Observer and Parity Space Approaches. *IEEE Transactions on Automatic Control*, 39(2):441–447, 1994.

[80] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[81] P. Miller, R.E. Swanson, and C.E. Heckler. Contribution Plot: A Missing Link in Multivariate Quality Control. *Applied Math & Computer Science*, 8(4):775–792, 1998.

[82] A. Naik. *Subspace Based Data-driven Designs of Fault Detection Systems*. PhD Dissertation, University of Duisburg-Essen, Germany, 2010.

[83] A. Naik, S. Yin, and S.X. Ding. Subspace Based Identification of Diagnostic Observer for Wiener Systems. In *Proceedings the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)*, pages 318–323, Barcelona, Spain, 2009.

[84] A. Naik, S. Yin, S.X. Ding, and P. Zhang. Recursive Identification Algorithms to Design Fault Detection Systems. *Journal of Process Control*, 20(8):957–965, 2010.

[85] I. Nimmo. Adequately Address Abnormal Situation Perations. *Chemical Engineering Progress*, 91(1):1361–1375, 1995.

[86] A. Nurkkala, F. Pettersson, and H. Saxen. Nonlinear Modeling Method Applied to Prediction of Hot Metal Silicon in The Ironmaking Blast Furnace. *Industrial & Engineering Chemistry Research*, 50(15):9236–9248, 2011.

[87] R.J. Patton and J. Chen. A Robust Parity Space Approach to Fault Diagnosis Based on Optimal Eigenstructure Assignment. In *International Conference on Control*, pages 1056–1062, 1991.

[88] R.J. Patton and J. Chen. Review of Parity Space Approaches to Fault Diagnosis for Aerospace Systems. *Journal of Guidance Control and Dynamics*, 17(2):278–285, 1994.

[89] R.J. Patton, P.M. Frank, and R.N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer Verlag, 2000.

[90] S.J. Qin. Partial Least Squares Regression for Recursive System Identification. In *Proceedings of the 32nd IEEE Conference on Decision and Control (CDC)*, pages 2617–2622, San Antonio, USA, 1993.

[91] S.J. Qin. Statistical Process Monitoring: Basics and Beyond. *Journal of Chemometrics*, 17(8-9):480–502, 2003.

[92] S.J. Qin. An Overview of Subspace Identification. *Computers & Chemical Engineering*, 30(10-12):1502–1513, 2006.

[93] S.J. Qin and Y.Y. Zheng. Quality-relevant and Process-relevant Fault Monitoring with Concurrent Projection to Latent Structures. *AIChE Journal*, 59(2):496–504, 2013.

[94] A. Rauzy. New Algorithms for Fault Trees Analysis. *Reliability Engineering & System Safety*, 40(3):203–211, 1993.

[95] W.H. Ray. *Advanced Process Control*. McGraw-Hill, 1981.

[96] R. Rengaswamy and V. Venkatasubramanian. A Syntactic Pattern Recognition Approach for Process Monitoring and Fault Diagnosis. *Engineering Applications of Artificial Intelligence*, 8(1):35–51, 1995.

[97] N.L. Ricker. Decentralized Control of The Tennessee Eastman Challenge Process. *Journal of Process Control*, 6(4):205–221, 1996.

[98] B.R. Robert and J. Antoni. Rolling Element Bearing Diagnostics: A Tutorial. *Mechanical Systems & Signal Processing*, 25(2):485–520, 2011.

[99] R. Rubini and U. Meneghetti. Application of The Envelope and Wavelet Transform Analysis for the Diagnosis of Incipient Faults in Ball Bearings. *Mechanical Systems & Signal Processing*, 15(2):287–302, 2001.

[100] E.L. Russell, L.H. Chiang, and R.D. Braatz. Fault Detection in Industrial Processes Using Canonical Variate Analysis and Dynamic Principal Component Analysis. *Chemometrics & Intelligent Laboratory Systems*, 51(1):81–93, 2000.

[101] H. Saxen and F. Pettersson. Nonlinear Prediction of The Hot Metal Silicon Content in The Blast Furnace. *ISIJ International*, 47(12):1732–1737, 2007.

[102] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[103] O. Slätteke. *Modeling and Control of the Paper Machine Drying Section*. PhD Dissertation, Lund University, Sweden, 2006.

[104] G. Stefatos and A.B. Hamza. Dynamic Independent Component Analysis Approach for Fault Detection and Diagnosis. *Expert Systems with Applications*, 37(12):8606–8617, 2010.

[105] N.Y. Steiner, D. Hissel, P. Mocoteguy, D. Candusso, D. Marra, C. Pianese, and M. Sorrentino. Application of Fault Tree Analysis to Fuel Cell Diagnosis. *Fuel Cells*, 12(2):302–309, 2012.

[106] J.Y. Tan, X.F. Chen, J.Y. Wang, H.X. Chen, H.R. Cao, Y.Y. Zi, and Z.J. He. Study of Frequency-shifted and Re-scaling Stochastic Resonance and Its Application to Fault Diagnosis. *Mechanical Systems & Signal Processing*, 23(3):811–822, 2009.

[107] N.F. Thornhill, S.C. Patwardhan, and S.L. Shah. A Continuous Stirred Tank Heater Simulation Model with Applications. *Journal of Process Control*, 18(3-4):347–360, 2008.

[108] R.J. Treasure, U. Kruger, and J.E. Cooper. Dynamic Multivariate Statistical Process Control Using Subspace Identification. *Journal of Process Control*, 14(3):279–292, 2004.

[109] E.M. Tzanakou. *Supervised and Unsupervised Pattern Recognition: Feature Extraction and Computational Intelligence*. CRC Press, 2000.

[110] V. Venkatasubramanian, R. Rengaswamy, and S.N. Kavuri. A Review of Process Fault Detection and Diagnosis: Part II: Qualitative Models and Search Strategies. *Computers & Chemical Engineering*, 27(3):313–326, 2003.

[111] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A Review of Process Fault Detection and Diagnosis: Part III: Process History Based Methods. *Computers & Chemical Engineering*, 27(3):327–346, 2003.

[112] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri. A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-Based Methods. *Computers & Chemical Engineering*, 27(3):293–311, 2003.

[113] D. Wang. Robust Data-driven Modeling Approach for Real-time Final Product Quality Prediction in Batch Process Operation. *IEEE Transactions on Industrial Informatics*, 7(2):337–371, 2011.

[114] J.A. Westerhuis, S.P. Gurden, and A.K. Smilde. Generalized Contribution Plots in Multivariate Statistical Process Monitoring. *Chemometrics & Intelligent Laboratory Systems*, 51(1):95–114, 2000.

[115] H. Wold. Causal Flows with Latent Variables: Partings of The Ways in The Light of NIPALS Modelling. *European Economic Review*, 3:67–86, 1974.

[116] S. Wold, K. Esbensen, and P. Geladi. Principal Component Analysis. *Chemometrics & Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.

[117] S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: A Basic Tool of Chemometrics. *Chemometrics & Intelligent Laboratory Systems*, 58(2):109–130, 2001.

[118] Y. Xiong and M. Saif. Robust Fault Detection and Isolation Via a Diagnostic Observer. *International Journal of Robust and Nonliear Control*, 10(14):1175–1192, 2000.

[119] F. Yang, D. Xiao, and S.L. Shah. Signed Directed Graph-based Hierarchical Modelling and Fault Propagation Analysis for Large-scale Systems. *Control Theory & Applications*, 7(4):537–550, 2013.

[120] S. Yin. *Data-driven Design of Fault Diagnosis Systems.* VDI Verlag, 2012.

[121] S. Yin, A. Naik, and S.X. Ding. Adaptive Process Monitoring Based on Parity Space Methods. In *Proceedings the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)*, pages 947–952, Barcelona, Spain, 2009.

[122] S. Yin, S.X. Ding, P. Zhang, A. Haghani, and A. Naik. Study on Modifications of PLS Approach for Process Monitoring. In *Proceedings of the 18th IFAC World Congress*, pages 12389–12394, Milano, Italy, 2011.

[123] S. Yin, S.X. Ding, A. Haghani, H.Y. Hao, and P. Zhang. A Comparison Study of Basic Data-driven Fault Diagnosis and Process Monitoring Methods on the Benchmark Tennessee Eastman Process. *Journal of Process Control*, 22(9):1567–1581, 2012.

[124] S. Yin, S.X. Ding, A. Haghani, and H.Y. Hao. Data-driven Monitoring for Stochastic Systems and Its Application on Batch Process. *International Journal of System Science*, 44(7):1366–1376, 2013.

[125] J. Yu, J.Y. Chen, and M.M. Rashid. Multiway Independent Component Analysis Mixture Model and Mutual Information Based Fault Detection and Diagnosis Approach of Multiphase Batch Processes. *AIChE Journal*, 59(8):2761–2779, 2013.

[126] M. Zasadzinski, H.S. Ali, H. Rafaralahy, and E. Magarotto. Disturbance Decoupled Diagnostic Observer for Singular Bilinear Systems. In *Proceedings of The 2001 American Control Conference*, pages 1455–1460, Virginia, USA, 2001.

[127] P. Zhang and S.X. Ding. Disturbance Decoupling in Fault Detection of Linear Periodic Systems. *Automatica*, 43(8):1410–1417, 2007.

[128] P. Zhang and S.X. Ding. On Monotonicity of a Class of Optimal Fault Detection Performance Versus Sampling Period. In *Proceeding of IEEE Conference on Decision and Control (CDC)*, pages 2596–2601, New Orleans, USA, 2007.

[129] P. Zhang and S.X. Ding. A Model-free Fault Detection Approach of Continuous-time Systems from Time Domain Data. *International Journal of Automation and Computing*, 4(2):189–194, 2007.

[130] P. Zhang and S.X. Ding. An Integrated Trade-off Design of Observer Based Fault Detection Systems. *Automatica*, 44(7):1886–1894, 2008.

[131] P. Zhang and S.X. Ding. Influence of Sampling Period on a Class of Optimal Fault Detection Performance. *IEEE Transactions on Automatic Control*, 54(6):1396–1402, 2009.

[132] P. Zhang and S.X. Ding. A Lifting Based Approach to Observer Based Fault Detection of Linear Periodic Systems. *IEEE Transactions on Automatic Control*, 57 (2):457–462, 2012.

[133] P. Zhang, S.X. Ding, G. Z. Wang, and D.H. Zhou. Fault Detection of Linear Discrete-time Periodic Systems. *IEEE Transactions on Automatic Control*, 50(2): 239–244, 2005.

[134] P. Zhang, T. Jeinsch, S.X. Ding, and P. Liu. Process Monitoring and Fault Diagnosis - Status and Applications. In *Proceedings of the 18th IFAC World Congress*, pages 12401–12406, Milano, Italy, 2011.

[135] D.G. Zheng. *System Identification and Model-based Control for A Class of Distributed Parameter Systems*. PhD Dissertation, Texas Tech University, USA, 2003.

[136] D.H. Zhou, G. Li, and S.J. Qin. Total Projection to Latent Structures for Process Monitoring. *AIChE Journal*, 56(1):168–178, 2010.

[137] J. Zhou, K. Chee, F. Lewis, and Z. Zhong. Intelligent Diagnosis and Prognosis of Tool Wear Using Dominant Feature Identification. *IEEE Transactions on Industrial Informatics*, 5(4):454–464, 2009.

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.