MODEL-BASED AND DATA DRIVEN FAULT DIAGNOSIS METHODS WITH APPLICATIONS TO PROCESS MONITORING

by

Qingsong Yang

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Thesis Advisor: Prof. Kenneth A. Loparo

Electrical Engineering and Computer Sciences

CASE WESTERN RESERVE UNIVERSITY

May 2004

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the dissertation of

Qingsong Yang

candidate for the Ph.D. degree *.

(signed) Kenneth A. Loparo
(chair of the committee)

Maurice L. Adams

Vira Chankong

Marc Buchner

(date) 3/2/2004

*We also certify that written approval has been obtained for any proprietary material contained therein.

To my parents Mingai and Shunlan,

for everything I am now,

and my wife Yunli,

for her love, support and understanding.

Contents

List of Tables

List of Figures

Acknowledgements

First and above all, I am really grateful to my advisor Prof. Kenneth Loparo. This work would not have been accomplished without his invaluable guidance and infinite patience. My presence at Case Western Reserve University is an enjoyable and memorable experience in my whole life mainly because I have a great advisor.

I also thank my committee members, Prof. Marc Buchner, Vira Chankong, and Maurice Adams, for their valuable help and suggestions.

I thank all the faculty, staff and students in the department, especially Amit Sinha, Jianmin He, Ying Wang and John Turnbull for their assistance, understanding and those interesting conversations.

Finally I appreciate the financial support from Case Western Reserve University, that made my education in the U.S. possible.

Model-based and Data Driven Fault Diagnosis Methods with Applications to Process Monitoring

Abstract

by

Qingsong Yang

This thesis discusses statistical model-based as well as data driven fault diagnosis approaches with applications to process monitoring.

A model-based technique, the Multiple Model Extended Kalman Filter (MMEKF), is proposed for the fault diagnosis of nonlinear stochastic systems. The MMEKF system consists of a bank of EKFs, where each filter is tuned to a specific fault. The residual, which is the difference between the measurement and the predicted measurement from the EKF, is an indicator of how well the filter is performing. Analysis of the residual can be used to evaluate the match between the actual measurement and the corresponding filter output, and therefore detect and isolate a possible fault. Hence, this multiple model approach has an inherent fault detection and isolation capability. A novel residual evaluation scheme combining a multivariate statistical technique, Principal Component Analysis (PCA), and an efficient search algorithm, is proposed to improve the fault evaluation process.

Simulation results on three different kinds of faults, actuator, sensor and process fault, show that the MMEKF fault diagnosis system can successfully perform the fault detection and isolation task. The fault magnitude can also be estimated by

PCA residual evaluation technique.

Process monitoring based on PCA has been a very popular topic in the context of data driven fault diagnosis techniques in recent years. However, two fundamental statistical assumptions have limited the performance of the conventional PCA approach for many industrial process control applications. Three variations of the conventional PCA, namely Adaptive PCA, Moving PCA (MPCA), and Multi-Scale PCA (MSPCA), are presented to increase the range of applicability of PCA. Each of these three extensions deals with a certain limitation of the conventional PCA. When the process is normal but undergoing a slow change which is still considered to be normal, Adaptive PCA is preferred. MPCA and MSPCA might be good approaches for detecting a slow process drift since they are more sensitive to small changes.

An industrial application for the Adaptive PCA and simulation studies for the MPCA and MSPCA show superior performance over the conventional PCA.

In addition, these approaches are also expected to improve the robustness of residual evaluation for the MMEKF approach when the effect of model uncertainty and unmodeled dynamics and disturbance cause the residuals to deviate from the assumed white noise characteristics of an optimal filter.

Chapter 1

Introduction

1.1 Literature Survey

Fault diagnosis has been becoming more and more important for process monitoring because of the increasing demand for higher performance as well as for increased safety and reliability of dynamic systems. Early diagnosis of process faults while the system is still operating in a controllable region can help avoid abnormal event progression and reduce productivity loss, which in turn can help avoid major system breakdowns and catastrophes. Hence, fault diagnosis is a major research topic attracting considerable interest from industrial practitioners as well as academic researchers.

There is an abundance of literature on process fault diagnosis ranging from analytical methods to artificial intelligence and statistical approaches. From a modelling perspective, there are methods that require accurate process models, semi-quantitative models, or qualitative models. At the other end of the spectrum, there are methods that do not assume any form of model information and rely only on historic process data. In addition, given the process knowledge, there are different search techniques

that can be applied to perform diagnosis. Fault diagnosis methods surveyed in [169], [170] and [171] can be classified into 2 general categories, model-based and data driven methods, based on the process knowledge that is required *a priori*. The hierarchy of fault diagnosis approaches is shown in Fig. 1.1.

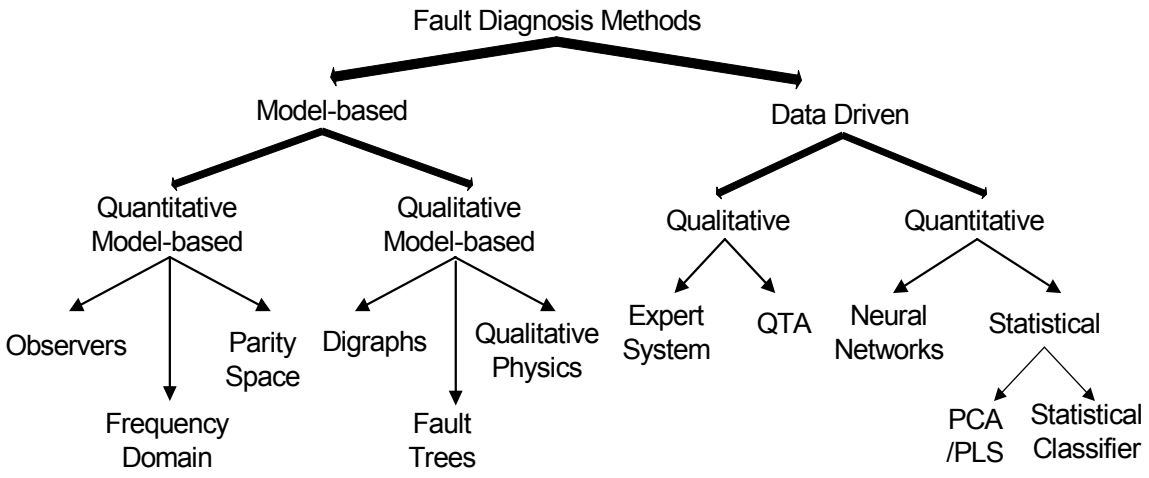

Figure 1.1: Classification of Fault Diagnosis Methods

The type of *a priori* process knowledge used is the most important distinguishing feature for classifying fault diagnosis systems. The basic *a priori* knowledge that is needed for fault diagnosis is the set of failures and the relationship between the observations (symptoms) and the failures. A diagnostic system may have them explicitly (as in a table lookup), or it may be inferred from some source of domain knowledge. The *a priori* domain knowledge may be developed from a fundamental understanding of the process using first-principles knowledge. Such knowledge is referred to as deep, causal or model-based knowledge in [105]. On the other hand, knowledge may be obtained from past experience with the process. This knowledge is referred to as

shallow, evidential or process history-based knowledge.

For model-based methods, the model-based *a priori* knowledge can be broadly classified as qualitative or quantitative. The model is usually developed based on some fundamental understanding of the physics of the process. In quantitative models this understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of the system. In qualitative model equations these relationships are expressed in terms of qualitative functions.

For data-driven approaches, only the availability of a large amount of historical process data is assumed. There are different ways in which this data can be transformed and presented as *a priori* knowledge to a diagnostic system. This is known as the feature extraction process from the process history data, and is done to facilitate later diagnosis. This extraction process can mainly proceed as either quantitative or qualitative feature extraction. In quantitative feature extraction one can perform either statistical or non-statistical feature extraction.

There might be some overlap between the model-based and data-driven approaches, this is just one classification based on whether or not the knowledge about process characteristics are required.

1.1.1 Model-based Fault Diagnosis Approaches

1.1.1.1 Quantitative Model-based Methods

Most of the work on quantitative model-based approaches have been based on using general input-output and state space models to generate residuals. These ap-

proaches can be classified into observer, parity space and frequency domain methods. Good survey papers include [56], [47], and [119]. The mathematical model-based approach adopted in this thesis falls into the observer category.

- **Observer or filter-based approaches:** The basic idea behind the observer or filter-based approaches is to estimate the outputs of the system from the measurements (or a subset of measurements) by using either observers in a deterministic setting ([10]; [25]; [44]; [45]; [122]; [121]) or statistical filters (e.g. the Kalman filter) in a stochastic setting ([104]; [182]; [195]; [143]). Then, the (weighted) output estimation errors (or innovations in the stochastic case) are used as the residuals. Depending on the circumstances, one may use linear ([115]) or nonlinear([44]; [146] [147]), full or reduced-order, fixed or adaptive observers (or Kalman filters) ([9]; [59]).

- **Parity space approaches:** Parity equations are rearranged and usually transformed variants of input-output or state space models of the plant ([55]; [55]). The basic idea is to check the parity (consistency) of the plant models with sensor outputs (measurements) and known process inputs. The idea of this approach is to rearrange the model structure so as to get the best fault isolation. Dynamic parity relations were introduced by Willsky ([182]). Redundancy provides freedom in the design of residual generating equations so that further fault isolation can be achieved. Fault isolation requires the ability to generate residual vectors which are orthogonal to each other for different faults. Gertler *et al.* ([53]; [52]) suggested a so-called "orthogonal parity equation" approach

in designing structured residual sets. The design of directional residual vectors using parity relations is not straightforward. The systematic approaches of designing parity equations with directional properties are presented in [54], and [52]. Chow and Willsky ([23]) proposed a procedure to generate parity equations from the state space representation of a dynamic system.

It has been observed by several researchers that some correspondence exists between observer-based and parity relation approaches. A full derivation of this structure equivalence can be found in [120].

- **Frequency domain (factorization) approaches:** Residuals can also be generated in the frequency domain via factorization of the transfer function of the monitored system. A comprehensive study was made by Ding and Frank [33], in which the FDI problem was systematically formulated and solved by factorization techniques. The recent developments including robustness issues can be found in [46]. An analogy between the observer-based residual generator and the factorization-based approach is shown in [18] for linear systems.

1.1.1.2 Qualitative Model-based Methods

Based on various forms of qualitative knowledge used in fault diagnosis, qualitative model-based approaches can be classified into digraphs, fault trees and qualitative physics methods.

- **Causal model approaches using digraphs:** Cause-effect relations or models can be represented in the form of signed digraphs (SDG). A digraph is a graph

with directed arcs between the nodes and SDG is a graph in which the directed arcs have a positive or negative sign attached to them. The directed arcs lead from the 'cause' nodes to the 'effect' nodes. SDGs provide a very efficient way of representing qualitative models graphically and have been the most widely used form of causal knowledge for process fault diagnosis. Iri *et al.* [65] were the first to use SDG for fault diagnosis. From SDG, they derive what is called a cause-effect graph (CE graph). Umeda *et al.* [158] showed how SDG can be obtained from differential algebraic equations for the process. Shiozaki *et al.* [148] addressed the issue of conditional arcs in their SDG representation. Shiozaki *et al.* [149] also extended the idea of SDG to include five-range patterns instead of the usual three-range pattern used in the standard SDG. Kokawa *et al.* [79] used partial system dynamics, statistical information about equipment failure, and digraphs to represent the failure propagation network for identifying fault location. Rule-based methods using SDG have been used for fault diagnosis by Kramer and Palowitch [80]. An important work in the field of steady-state qualitative simulation (QSIM) using SDG has been presented by Oyeleye and Kramer [117].

In recent years, Wilcox and Himmelblau [180] [181] have approached the problem of fault diagnosis using what is called possible cause and effect graph (PCEG) models. Vaidhyanathan and Venkatasubramanian [160] have used digraph-based models for automated HAZOP analysis. Use of SDGs for multiple fault detection is demonstrated by Vedam and Venkatasubramanian [164].

Improvement of fault resolution in SDG models through the use of fuzzy set theory is discussed by Han *et al.* [61]. Genovesi *et al.* [51] have presented a framework for process supervision using fuzzy logic-based fault diagnosis. Li and Wang [88] have presented how fuzzy digraphs can be used for qualitative and quantitative simulation of the temporal behavior of process systems.

- **Fault trees approaches:** Fault trees are used in analyzing system reliability and safety. Fault tree analysis was originally developed at Bell Telephone Laboratories in 1961. Fault tree is a logic tree that propagates primary events or faults to the top level event or a hazard. The tree usually has layers of nodes. At each node different logic operations like AND and OR are performed for propagation. Fault-trees have been used in a variety of risk assessment and reliability analysis studies by Kelly and Lees [78], and Ulerich and Powers [157]. Interested readers should refer to [49] and [85] for a more detailed description of fault tree analysis.

- **Qualitative physics approaches:** Qualitative physics knowledge in fault diagnosis has been represented in mainly two ways. The first approach is to derive qualitative equations from the differential equations termed as confluence equations. Considerable work has been done in this area of qualitative modelling of systems and representation of causal knowledge, including [150], [30] and [66].

  The other approach in qualitative physics is the derivation of qualitative behavior from the ordinary differential equations (ODEs). These qualitative behaviors

for different failures can be used as a knowledge source. Sacks [142] examines piece-wise linear approximations of nonlinear differential equations through the use of a qualitative mathematical reasoner to deduce the qualitative properties of the system. Kuipers [84] predicts qualitative behavior by using qualitative differential equations (QDEs) that are an abstraction of the ODEs that represent the state of the system. In terms of applications of qualitative models in fault diagnosis, qualitative simulation (QSIM) and qualitative process theory (QPT) have been the popular approaches. Examples of research work in QSIM include [130], [102], and [76]. Examples of using the QPT framework in process fault diagnosis include [57], [58], and [41].

1.1.2 Data-driven Fault Diagnosis Approaches

1.1.2.1 Quantitative Data-driven Methods

Methods that extract quantitative information can be broadly classified as non-statistical or statistical methods. Neural networks are an important class of non-statistical classifiers. Principal component analysis (PCA)/partial least squares (PLS) and statistical pattern classifiers form a major component of the statistical feature extraction methods.

- **Multivariate statistical approaches:** The successful applications of multivariate statistical methods to fault diagnosis such as Principal Component Analysis (PCA) and Partial Least Squares (PLS) have been extensively reported in the literature. Overviews of using PCA and PLS in process analysis

and control, fault diagnosis were given by MacGregor *et al.* [81] [95], and Wise and Gallagher [185].

In earlier work, Kresta *et al.* [81] presented the basic methodology of using the multivariate SPC procedure to handle large numbers of process and quality variables for continuous processes. Later on, Nomikos and MacGregor [112] extended the use of multivariate projection methods to batch processes by using multiway PCA. To deal with nonlinearity, Qin and McAvoy [126] proposed a neural net PLS approach that incorporated feedforward networks into the PLS modelling. In order to handle nonlinearity in batch processes, Dong and McAvoy [34] utilized a nonlinear PCA method. Raich and Cinar [128] [129] proposed an integral statistical methodology combining PCA and discriminate analysis techniques, using distance and angle based discriminants.

Chemometrics is defined as the science of relating measurements made on a chemical system to the state of the system via application of mathematical or statistical methods. Chemometric techniques have been applied to chemical engineering processes ([72]; [123]) in recent years.

Dunia *et al.* [39], Qin and Li [124] used PCA for sensor fault detection, identification and reconstruction. Dunia and Qin [38] looked at PCA from a geometric point of view and presented a methodology that analyzed the fault subspace for process and sensor fault detection.

A major limitation of conventional PCA monitoring is that the PCA model

is time invariant, while most real processes are time-varying. Hence the PCA model should also be recursively updated. An adaptive monitoring approach using recursive PLS has been presented by Qin [125], and a similar recursive PCA approach was proposed by Li *et al.* [89]. Another promising variant of the PCA approach is the multiscale PCA (MSPCA) approach which integrates PCA and wavelet analysis ([3]; [93]; [192]; [2]).

- **Statistical classifier approaches:** Fault diagnosis is essentially a classification problem and hence can be cast in a classical statistical pattern recognition framework. Fault diagnosis can be considered as a problem of combining, over time, the instantaneous estimates of the classifier using knowledge about the statistical properties of the failure modes of the system ([138]; [114]).

- **Neural networks approaches:** Considerable interest has been shown in the literature to the application of neural networks for fault diagnosis. In general, neural networks that have been used for fault diagnosis can be classified along two dimensions: (i) the architecture of the network such as sigmoidal, radial basis and so on; and (ii) the learning strategy such as supervised and unsupervised learning.

  The most popular supervised learning strategy in neural networks has been the back-propagation algorithm. There are a number of papers that address the problem of fault diagnosis using back-propagation neural networks. In chemical engineering, Watanabe *et al.* [177], Venkatasubramanian and Chan [168],

Ungar *et al.* [159] and Hoskins *et al.* [64] were among the first researchers to demonstrate the usefulness of neural networks for fault diagnosis. A detailed and thorough analysis of neural networks for fault diagnosis in steady-state processes was presented by Venkatasubramanian *et al.* [172]. This work was later extended to utilize dynamic process data by Vaidyanathan and Venkatasubramanian [161]. A hierarchical neural network architecture for the detection of multiple faults was proposed by Watanabe *et al.* [176].

Most of the work on improvement of performance of standard back-propagation neural networks for fault diagnosis is based on the idea of explicit feature presentation to the neural networks by Fan *et al.* [42], Farell and Roat [43], Tsai and Chang [155], and Maki and Loparo [97]. Modifications to the selection of basis functions have also been suggested to the standard back-propagation network. For example, Leonard and Kramer [86] suggested the use of radial basis function networks for fault diagnosis applications. Kavuri and Venkatasubramanian ([73]; [74]; [75]) generalized radial units to Gaussian units and proposed methods to solve the hidden node problem.

Different network architectures have been used for the problem of fault diagnosis. For example, Bakshi and Stephanopoulos [4] proposed Wavenet: a multi-resolution hierarchical neural network. Self-organizing neural network structures such as the ART2 network ([14]) have also been extensively used in fault diagnosis. Whiteley and Davis [179] demonstrate the use of the ART2 network for the interpretation of sensor data. Chen *et al.* [16] and Wang *et al.*

[175] discuss the integration of wavelets with ART networks for the development of diagnostic systems.

1.1.2.2 Qualitative Data-driven Methods

Two of the major methods that extract qualitative history information are expert systems and qualitative trend analysis (QTA).

- **Expert system approaches:** Rule-based feature extraction has been widely used in expert systems for many applications. An expert system is generally a very specialized system that solves problems in a narrow domain of expertise. Initial attempts at the application of expert systems for fault diagnosis can be found in the work of Henley [62], and Niida [109]. Structuring the knowledge-base through hierarchical classification can be found in [132]. Ideas on knowledge-based diagnostic systems based on the task framework can be found in [131]. A rule-based expert system for fault diagnosis in a cracker unit is described in [167]. More work on expert systems in chemical process fault diagnosis can be found in [127] and [141].

  A number of other researchers have also worked on the application of expert systems to diagnostic problems. Basila *et al.* [5] have developed a supervisory expert system that uses object-based knowledge representation to represent heuristic and model-based knowledge. Zhang and Roberts [193] have presented a methodology for formulating diagnostic rules from the knowledge of system structures and component functions. Becraft and Lee [11] have proposed an

integrated framework comprising of a neural network and an expert system. Tarifa and Scenna [152] have proposed a hybrid system that uses signed directed graphs (SDG) and fuzzy logic. Zhao *et al.* [194] have presented a wavelet-sigmoid basis neural network and expert system based integrated framework for fault diagnosis of a hydrocracking process. Wo *et al.* [187] have presented an expert fault diagnostic system that uses rules with certainty factors. Leung and Romagnoli [87] have presented a probabilistic model-based expert system for fault diagnosis. An expert system approach for fault diagnosis in batch processes is discussed in Scenna [145].

- **Qualitative trend analysis approaches:** Trend analysis and prediction are important components of process monitoring and supervisory control. Trend modelling can be used to explain the various important events that happen in a process, to diagnosis malfunctions and to predict future states. Cheung and Stephanopoulos [21] have built a formal framework for the representation of process trends. Janusz and Venkatasubramanian [69] identify a comprehensive set of primitives by which any trend can be represented. Rengaswamy and Venkatasubramanian [137] have shown how primitives can be extracted from raw noisy sensor data by treating the problem of primitive identification as a classification problem using neural networks. Vedam and Venkatasubramanian [165] proposed a wavelet theory based adaptive trend analysis framework and later proposed a dyadic B-Splines based trend analysis algorithm [166]. Recently, Rengaswamy *et al.* [136] have discussed the utility of trend modelling in

control loop performance assessment.

1.2 Thesis Summary

This thesis discusses statistical model-based as well as data driven fault diagnosis approaches applied to process monitoring.

A model-based technique, the Multiple Model Extended Kalman Filter (MMEKF), is proposed for the fault diagnosis of nonlinear stochastic systems. The MMEKF system consists of a bank of EKFs, where each filter is tuned to a specific fault. The residual, which is the difference between the measurement and the predicted measurement from the EKF, is an indicator of how well the filter is performing. Since each Kalman filter is sensitive to a particular fault, a residual with known statistical properties would suggest a match between the actual measurement and the corresponding filter, and therefore a possible fault, so characteristics of the residuals from various filters gives a relative indication of how adequately each of these models represents the actual fault status. Hence, this multiple model approach has an inherent fault detection and isolation capability.

Over the past two decades, the growing demand for reliability and production sustainability in industrial processes has urged increasing research effort on the development of Fault Diagnosis and Accommodation (FDA) or the so-called Fault Tolerant Control (FTC). The FDA or FTC strategies are capable of tolerating component malfunctions whilst still maintaining control objectives through control system reconfiguration or the use of control laws that are insensitive to some faults. The knowledge

of the fault information is very important for fault tolerant control. Hence, the ability to know not only which fault occurs but the size of the fault is essential. A novel residual evaluation scheme combining a multivariate statistical technique, Principal Component Analysis (PCA), and an efficient search algorithm, is proposed to achieve fault evaluation.

Simulation results on three different kinds of faults, actuator, sensor and process fault, show that the MMEKF fault diagnosis system can successfully perform fault detection and isolation task. The fault magnitude can also be estimated by the PCA residual evaluation technique.

Process monitoring based on PCA has been a very popular topic in the context of data driven fault diagnosis techniques in recent years. However, two fundamental statistical assumptions, measured variables are independent and identically distributed normal random variables; measured variables are uncorrelated in time (not auto-correlated), have limited the performance of the conventional PCA approach for industrial process control applications. Three variations of the conventional PCA, namely Adaptive PCA, Moving PCA (MPCA), and Multi-Scale PCA (MSPCA), are presented. Each of these three extensions deals with a certain limitation of the conventional PCA. When the process is normal but undergoing a slow change which is still considered to be normal, Adaptive PCA is preferred. MPCA and MSPCA might be good approaches for detecting slow process drift since they are more sensitive to small change.

An industrial application for Adaptive PCA and simulation studies for MPCA

and MSPCA show superior performance over conventional PCA. In addition, these approaches are expected to improve the robustness of residual evaluation for the MMEKF approach when the effect of model uncertainty and unmodeled dynamics and disturbance cause the residuals to deviate from the assumed white noise characteristics of an optimal filter.

1.3 Thesis Organization

The thesis is organized as follows:

Chapter 1 is the introduction. It gives a summary of the previous work reported in the literature and an overview of the topics discussed in the thesis.

Chapter 2 presents a model-based fault diagnosis scheme, the multiple model extended Kalman filter. Basic principles of the model-based fault diagnosis are discussed, followed by a detailed description of modelling of the faults studied in this thesis. Theoretical foundation of Kalman filter is then given before the MMEKF fault diagnosis system design. PCA preliminaries are introduced to facilitate the understanding of the mechanism of the PCA residual evaluation approach. A simulation study based on the well-known CSTR (Continuously well-Stirred Tank Reactor) model is conducted to evaluate the MMEKF fault diagnosis performance.

Chapter 3 discusses a multivariate statistical data driven fault diagnosis approach, PCA. Three variations of conventional PCA, Adaptive PCA, Moving PCA, and Multi-scale PCA, are also presented. The methodology and detailed process monitoring scheme are shown for each technique. An industrial application for Adaptive

PCA and simulation studies for MPCA and MSPCA complete this chapter.

Chapter 4 gives concluding remarks, summarizes the main contributions of the thesis and discusses directions for future work.

Chapter 2

Model-Based Fault Diagnosis Approach: Multiple Model Extended Kalman Filter

The model-based approach for process fault diagnosis has been extensively studied over the last three decades, both in a research context and also in the domain of industrial applications. A great variety of methods have been proposed in the literature, based on the mathematical models of the monitored processes and the techniques developed in modern control theory. It is not the purpose of this chapter to provide a comprehensive literature survey about the model-based fault diagnosis, but rather a brief introduction of the basics in the first section. Then a state-estimator-based scheme, namely the multiple model extended Kalman filter, is combined with the multivariate statistical signal processing technique, principal component analysis, to provide a novel fault diagnosis solution. Analysis and discussion of the simulation results of a highly nonlinear plant model conclude the chapter.

2.1 Basic Principles of the Model-Based Fault Diagnosis

Model-based fault diagnosis can be deemed as a monitoring procedure of FDI (fault detection and isolation) in a system by comparing the system's available measurements with *a priori* information represented by the system's mathematical model. The procedures of using model information to generate additional signals, to be compared with the original measurement quantities are known as *analytical redundancy* approaches, which are considered to be more reliable and cost effective than the alternative approach known as *hardware redundancy* ([163]). Analytical redundancy is a form of *dissimilar redundancy* ([119]) in the sense that given one measurement from the dynamic system under control, a mathematical model of the system is used to generate estimates of other measurements, thereby developing redundancy in an analytical form. Fig. 2.1 illustrates the hardware and analytical redundancy concepts.

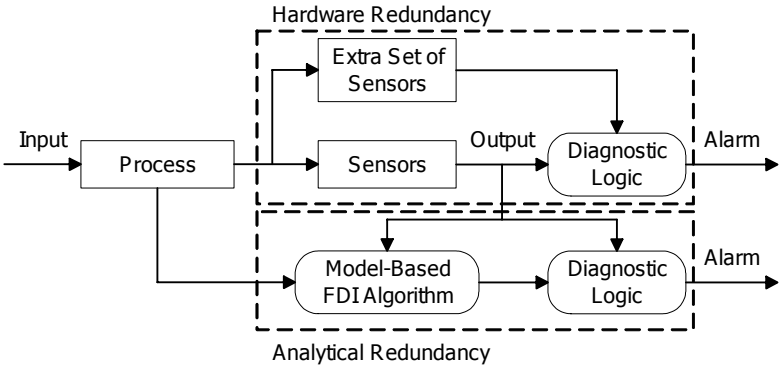

Figure 2.1: Hardware vs Analytical Redundancy

According to generally accepted terminology, fault diagnosis consists of the fol-

lowing tasks ([56]):

- **Fault detection:** to make a binary decision - either that something is going wrong or that everything is fine.
- **Fault isolation:** to determine the location of the fault, e.g., which sensor or actuator has become faulty.
- **Fault evaluation:** to estimate the size and type or nature of the fault.

The relative importance of these three tasks are obviously subjective, however detection is an absolute must for any practical system and isolation is almost equally as important. Fault evaluation, on the other hand, may not be essential if no reconfiguration action is involved. Hence, fault diagnosis is very often considered as fault detection and isolation, abbreviated as FDI, in the literature.

In the mean time, fault prognosis, e.g. the process of predicting the future condition of a system based on the current diagnostic state of the system and its current operating and failure history data, is becoming increasingly important for continuous-time production processes, power plants, and mechanical systems where structure durability and operational reliability are critical. The prognostic method intelligently utilizes diagnostic results, experience based information and statistically estimated future conditions to determine the remaining useful life or failure probability of a component or subsystem. To reduce the production loss appreciably, it is vital to have advance warning of impending failures even up to several months ahead, so that the necessary planning can be made.

The focus of this thesis is on fault diagnosis.

2.1.1 General Structure of Model-Based Fault Diagnosis Systems

The general procedure of FDI looks like this: faults are first detected by setting a (fixed or variable) threshold on a residual quantity generated from the difference between real measurements and estimates of these measurements using a mathematical model. A number of residuals can be designed with each tuned to be sensitive to individual faults occurring in different locations in the system. The subsequent analysis of each residual, once a threshold is exceeded, then leads to fault isolation.

Fig. 2.2 illustrates the general and conceptual structure of a model-based fault diagnosis system comprising the two stages of *residual generation* and *decision making* ([18]). This two-stage structure was first suggested in [22] and now is widely accepted by the fault diagnosis community. These two stages are described as follows:

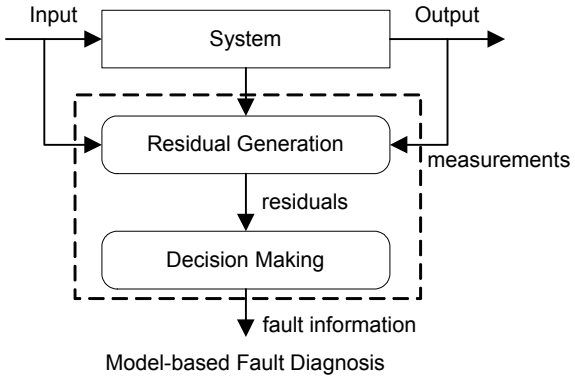

Figure 2.2: General Structure of Model-Based Fault Diagnosis Systems

1. **Residual Generation:**

   Its purpose is to generate a fault index signal – residual, using available input and output information from the monitored system. The sensory measurements are compared to analytically obtained values of the respective variable. Such computations use present and/or previous measurements of variables and a mathematical model that describes their relationship. The idea can be extended to the comparison of analytically generated quantities obtained through different computations. In either case, the resulting differences are called *residuals*. The residual should normally be a zero-mean or close to zero-mean noise sequence in the stochastic system when no fault is present, but is distinguishably different from zero-mean when a fault occurs. The algorithm (or processor) used to generate the residuals is called a *residual generator*. Residual generation is thus a procedure for extracting fault symptoms from the system, with the specific fault symptom of interest represented by the residual signal. The residuals should ideally carry only fault information and are independent of the system operating state.

   In practice, the most frequently used FDI approaches use information known *a priori* about the characteristics of monitored signals. The main shortcomings of this group of methods are: (a) the necessity to have *a priori* information about the characteristics of the signals, (b) the inevitable dependence of these characteristics on operating states of the system which are not known *a priori* and can change during the course of system operation. The introduction of

residuals provides the model-based FDI approaches with a great advantage over traditional methods simply because the residuals are independent of the system operating state and responsive to faults in characteristic or predictive manners. Based on the mathematical model, many invariant relations (dynamic or static) among different system variables can be derived and used to design the residuals. The residual generation can be interpreted in terms of the redundant signal structure as illustrated in Fig. 2.3 ([6]).

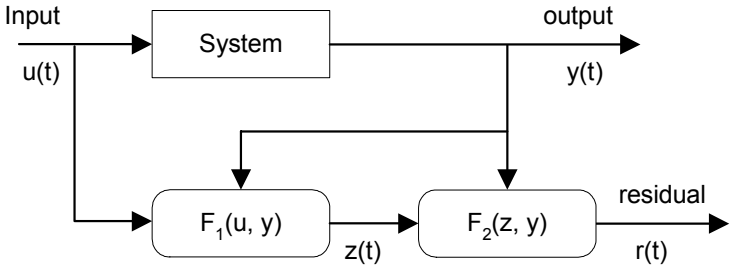

Figure 2.3: Redundant Signal Structure in Residual Generation

In this structure, the system (processor or algorithm) $F_1(u, y)$ generates an auxiliary (redundant) signal $z(t)$ which, together with $y(t)$ generates the residual $r(t)$ that satisfy the following invariant relation:

$$\overline{r}(t) = F_2(z(t), y(t)) = 0 \tag{2.1}$$

for the fault-free case. When any fault occurs in the system, this invariant relation will be violated and the residual will be non-zero mean.

The simplest approach to residual generation is using system duplication, i.e. the system F_1 is constructed to be identical to the original system model and has

the same input signal as the system. In this case, the signal $y(t)$ is not required in the system block F_1 which is then simply a system simulator. This, of course, requires that the internal states and the initial conditions of the internal states are known *a priori*. $z(t)$ is the simulated output of the system, and the residual is the difference between $z(t)$ and $y(t)$. The disadvantage of this method is that the stability of the simulator can not be guaranteed when the system being monitored is unstable, as a consequence of the open-loop system model in FDI (although it is under feedback control) (see section 2.2.3).

A direct extension to the simulator-based residual generation is to replace the simulator with an output estimator which requires both system input and output information (see Fig. 2.4). Now the system uses both signals $u(t)$ and $y(t)$ to generate an estimation of a linear function of the output $y(t)$, say $My(t)$, and the system F_2 can be defined as $F_2(z, y) = Q(z - My)$ with Q as a static (or dynamic) weighting matrix.

No matter what type of method is used, a residual generator is just a processor whose input consists of both input and output of the system being monitored.

2. **Decision Making:**

The residuals are examined for the likelihood of faults, and a decision rule is then applied to determine if any faults have occurred and which fault(s) have occurred. A decision process may consists of a simple threshold test on the instantaneous values or moving averages of the residuals, or comparing the

residuals to a set of patterns (signatures) known to belong to simple faults, or by the use of more complex logical procedures like statistical decision theory, e.g., generalized likelihood ratio (GLR) testing or sequential probability ratio testing (SPRT) ([184]; [6]; [8]; [26]).

2.1.2 Residual Generation Techniques

Most of the work in the field of quantitative model-based fault diagnosis is focused on the residual generation problem because decision making based on well designed residuals is relatively easy. A wide variety of methods are available for residual generation and they can be classified into the following categories:

1. **Observer-Based Approaches:**

   An example of the observer-based residual generator is illustrated by the block diagram of a linear full order observer in Fig. 2.4, where *f(t)* denotes the vector of faults to be detected, and *d(t)* denotes the vector of unknown inputs (disturbance, noise, modelling errors).

   According to the definition, the residual *r(t)* under the deterministic model is designed to be zero for the fault-free case and nonzero for faulty cases, i.e.

$$r(t) = 0 \quad iff \quad f(t) = 0 \tag{2.2}$$

   and

$$\begin{aligned} r(t) &\neq 0 \ \ if \ \ f(t) \neq 0 \text{ for fault detection} \\ r_i(t) &\neq 0 \ \ if \ \ f_i(t) \neq 0 \text{ for fault isolation} \end{aligned} \tag{2.3}$$

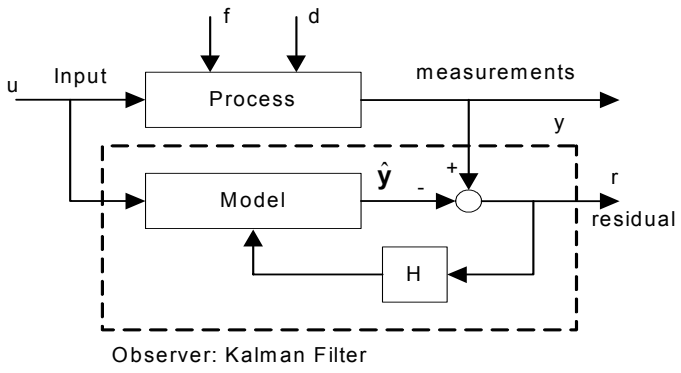

Figure 2.4: Full Order Observer for Residual Generation

where the $f_i(t)$ represent different faults to be isolated and $r_i(t)$ is the corresponding residual.

While residuals are zero under ideal conditions, in practical and stochastic situations, this is never true. Therefore, a threshold needs to be set to evaluate whether or not the residuals are under faulty condition. A fault can be detected by comparing the residual evaluation function $J(r(t))$ with a threshold function $T(t)$ according to the test given below:

$$\begin{cases} J(r(t)) \leqslant T(t) & for \quad f(t) = 0 \\ J(r(t)) > T(t) & for \quad f(t) \neq 0 \end{cases} \tag{2.4}$$

If the test is positive (i.e. the threshold is exceeded by the residual evaluation function), a fault is likely occurring. There are many ways of defining residual evaluation functions and determining thresholds.

An observer-based residual generation technique is used in the thesis study. A more detailed description will be found in the following section.

2. **Parity Space Approach:**

In the early development of fault diagnosis, the *parity vector* approach was applied to *static* or *parallel redundancy* schemes ([28]; [31]) which may be obtained directly from measurements or from analytical relations. A survey of these schemes can be found in [134]. There are typically two cases for arranging hardware redundancy, one is the use of sensors having identical or similar functions to measure the same variable, another is the use of dissimilar sensors to measure different variables but with their outputs being relative to each other. The basic idea of the parity vector approach is to provide a proper check of the parity (consistency) of the measurements of the monitored system. A general problem of the measurement of an n-dimensional vector using m sensors is presented to describe this approach. The measurement (algebraic) equation is:

$$y(k) = Cx(k) + f(k) + \xi(k) \tag{2.5}$$

where $y(k) \in \Re^m$ is the measurement vector, $x(k) \in \Re^n$ is the state vector, and $f(k)$ is the vector of sensor faults, $\xi(k)$ is a noise vector and C is an $m \times n$ measurement matrix.

With hardware (direct) redundancy, the dimension of $y(k)$ is larger than the dimension of $x(k)$.

$$m > n; \quad and \quad rank(C) = n$$

For such a system configuration, the number of measurements is greater than the number of variables to be sensed. Inconsistency in the measurement data is

then a metric that can be used initially for detecting faults and, subsequently for fault isolation.

For FDI purposes, the vector $y(k)$ can be combined into a set of linearly independent parity equations to generate the parity vector (residual):

$$r(k) = Vy(k) \tag{2.6}$$

The residual generation scheme based on direct redundant measurements is shown in Fig. 2.5.

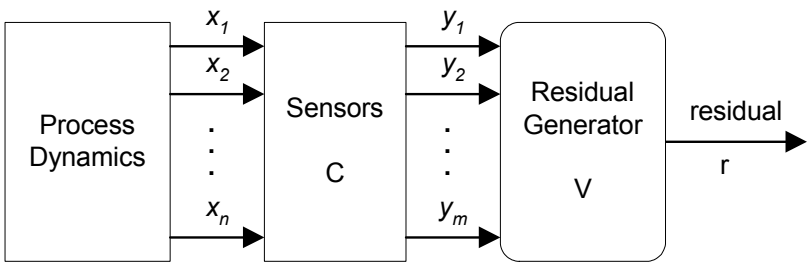

Figure 2.5: Residual Generation via Parallel Redundancy

In order to make $r(k)$ satisfy the usual requirement for a residual (zero value for the fault-free case), the parity matrix V must satisfy the condition:

$$VC = 0 \tag{2.7}$$

When this condition holds true, the residual (parity vector) only contains information on the faults and noise:

$$r(k) = v_1[f_1(k) + \xi_1(k)] + \cdots + v_m[f_m(k) + \xi_m(k)] \tag{2.8}$$

where v_i is the i_{th} column of V and $f_i(k)$ is the i_{th} element of $f(k)$ which models the fault in the i_{th} sensor.

Eq. 2.8 shows that the parity vector only contains information due to faults and noise (uncertainty), and is independent of the unmeasured state $x(k)$. Eq. 2.8 also reveals that the parity space (or residual space) is spanned by the columns of V, i.e. the columns of V form a basis for this space. Moreover, an interesting property can also be exploited: a fault in the i_{th} sensor implies a growth of the residual $r(k)$ in the direction of v_i. This ensures that a fault in the i_{th} sensor implies an increase in the norm of $r(k)$ in the direction of v_i. The *span* $\{V\}$ is called a *parity space*.

Using the notation of [28], a *fault detection decision function* is defined as:

$$DFD(k) = r(k)^T r(k) \tag{2.9}$$

If a fault occurs in the sensors, $DFD(k)$ will be greater than an predetermined threshold.

The *fault isolation decision function* is then:

$$DFI_i(k) = v_i^T r(k); \quad i \in \{1, 2, \cdots, m\} \tag{2.10}$$

For a given $r(k)$, a malfunctioning sensor is identified by computing the m value of $DFI_i(k)$. If $DFI_j(k)$ is the largest one of these values, the j_{th} sensor is the one which is most likely to have become faulty.

From the parity space point of view, the columns of V define m distinct *fault*

signature directions $(v_i, i = 1, 2, \cdots, m)$. Indeed, the fault isolation function $DFI_i(k)$ is a measure of the correlation of the parity (residual) vector with the fault signature directions. In order to isolate faults reliably, the generalized angles between the fault signature directions should be "as large as possible", i.e., to make $v_i^T v_j$ "as small as possible". Thus, optimal fault isolation performance will be achieved when the v_i satisfy the following relations:

$$\begin{cases} \min\{v_i^T v_j\};\ i \neq j,\ i, j \in \{1, 2, \cdots, m\} \\ \max\{v_i^T v_i\};\quad i \in \{1, 2, \cdots, m\} \end{cases} \tag{2.11}$$

The question that remains is how to design the matrix V. The traditional sub-optimal solution is to make ([134]):

$$VV^T = I_{m-n} \tag{2.12}$$

A further consequence of conditions 2.7 and 2.12 is that:

$$V^T V = I_m - C(C^T C)^{-1} C^T \tag{2.13}$$

The condition for the existence of a solution V for Eq. 2.7 is that $rank(C) = n < m$. This implies that the rows of C are linearly dependent, i.e., the outputs of the sensors are related by a static relation. This is where the terminology hardware redundancy comes from.

For the case $rank(C) = m < n$, a *direct redundancy* relation does not exist. However, "*temporal redundancy*" or "*serial redundancy*" relations may be constructed by collecting sensor outputs over a time interval (time window) (say,

$\{y(k-s), y(k-s+1), \cdots, y(k)\}$). A dynamic model of the system must be known and used in this case as the redundancy relationship is time dependent.

The parity relation approach can also be used to design a structured residual set for fault isolation ([101]; [52]). The design for isolating sensor faults is very straightforward. If c_i^T (the i_{th} row of C) and y_i (the i_{th} component of y) are used instead of C and y, the parity relation will only contain the output of the i_{th} sensor together with all inputs. This form of parity relation has been called a *single-sensor parity relation (SSPR)* ([101]) and the residual generated by this relation is only sensitive to a fault in the i_{th} sensor. For the actuator isolation problem, the structured residual set is more difficult to design. Massoumnia and Vander Velde studied this problem and pointed out that the isolation of actuator faults is not always possible ([101]). This conclusion is consistent with the observer-based approach.

3. **Factorization (Frequency Domain) Method:**

   Residuals can also be generated in the frequency domain via factorization of the transfer function of the monitored system. A comprehensive study was made by [33], in which the FDI problem was systematically formulated and solved by factorization techniques. The recent developments including robustness issues can be found in [46].

   This approach is based on the fact that any $m \times r$ proper rational transfer

function $G_u(s)$ can be factorized as ([173]):

$$G_u(s) = \tilde{M}^{-1}(s)\tilde{N}(s) \tag{2.14}$$

where $\tilde{M}(s)$ and $\tilde{N}(s)$ are stable, rational and realizable transfer functions.

Consider a linear system with the state space model given by:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + R_1 f(t) \\ y(t) = Cx(t) + Du(t) + R_2 f(t) \end{cases} \tag{2.15}$$

where $x \in \Re^n$ is the state vector, $u \in \Re^r$ is the input vector to the actuator and $y \in \Re^m$ is the measured system output vector; A, B, C and D are known system matrices with appropriate dimensions; $f(t) \in \Re^g$ is a fault vector, each element $f_i(t)$ $(i = 1, 2, \cdots, g)$ corresponds to a specific fault. The matrices R_1 and R_2 are known as fault entry matrices which model the effect of faults on the system. Both the input vector u and the output vector y are assumed to be measurable for FDI purposes.

An input-output transfer function representation for the system with possible faults is then described as:

$$y(s) = G_u(s)u(s) + G_f(s)f(s) \tag{2.16}$$

where

$$\begin{cases} G_u(s) = C(sI - A)^{-1}B + D \\ G_f(s) = C(sI - A)^{-1}R_1 + R_2 \end{cases} \tag{2.17}$$

Based on the factorization of $G_u(s)$ (Eq. 2.14), a residual generator can be designed as:

$$r(s) = Q(s)[\tilde{M}(s)y(s) - \tilde{N}(s)u(s)] \tag{2.18}$$

where $Q(s)$ denotes a dynamic residual weighting matrix. Substituting Eq. 2.16 into Eq. 2.18 together with Eq. 2.14, the residual becomes:

$$r(s) = Q(s)\tilde{M}(s)G_f(s)f(s) \tag{2.19}$$

which is only affected by the fault. The weighting factor Q(s) can be used to improve the residual performance responding to faults in a particular frequency region.

2.1.3 Fault Diagnosis for Stochastic Systems

While residuals are zero for deterministic systems under ideal conditions, in practice, this is never the case. Noise and disturbances are always present in the system and consequently, FDI for stochastic systems requires special treatment.

In order to suppress the noise effects on the residuals, the residual generator has to be specially designed to deal with the noise. A common approach for linear stochastic systems is to use Kalman-filter-based residual generators. While having a similar structure to the observer, approaches based on the Kalman filter have a residual generation mechanism derived from a stochastic model of the dynamic system. The following well-known fact is the foundation of all those approaches: under normal operation the Kalman filter residual (or innovations) vector (the difference

between the measurements and their Kalman filter estimates), is a zero-mean white noise process with known covariance matrix. [104] first proposed the use of different statistical tests on the innovations to detect faults in the system. Many variants of the idea of hypothesis testing for FDI have been published since then ([182]; [6]; [111]; [7]). The idea which is common to all these approaches is to test, among all possible hypotheses, that the system has a fault or is fault-free. As each fault type has its own signature, a set of hypotheses can be used and checked for the likelihood that a particular fault has occurred.

Some Kalman-filter-based state estimators are developed especially for FDI of stochastic systems, e.g.:

- Multiple Model Filters (MMFs):

  The basic idea is to design a bank of filters, with each model tuned to be sensitive to a particular failure, thus particular failures manifest themselves as residuals which remain in a fixed direction or under certain thresholds. Some of the examples include, multiple hypothesis filter-detectors in [183], a bank of Kalman filters for sensor failure detection in [27], a multiple model nonlinear filtering approach for finite state random jump processes ([92]; [91]; [90]), a multiple model adaptive estimation (MMAE) algorithm ([103]; [40]) for small perturbations of a nonlinear system using linearized system models.

- Two-Stage Bias-Correction Filters ([48]; [19]).

The statistical testing of the residuals for stochastic systems has the following

forms:

- Weighted sum-squared residual (WSSR) testing ([156]).
- χ^2 testing ([182]; [26]; [17]).
- Sequential probability ratio testing (SPRT) ([182]; [156]) and modified SPRT ([151]; [156]).
- Generalized likelihood ratio (GLR) testing ([184]; [77]).
- Cumulative sum algorithm ([110]).
- Multiple hypothesis testing ([13]).

2.1.4 Fault Modelling (Sensor, Actuator & Process Faults)

Model-based FDI is concerned mainly with on-line fault diagnosis, in which the diagnosis is carried out during system operation. This is because the system input and output information that is required by model-based FDI is only available when the system is in operation. One of the important concerns in the design of the FDI scheme is that it should not interfere with the control loop. The relationship between fault diagnosis (or supervision) and the control loop is shown in Fig. 2.6.

The information used for FDI is the measured output from the sensors and the input to the actuators. From Fig. 2.6, it is easy to see that the system model required in model-based FDI is the open-loop system model although we consider that the system is in the feedback control loop. The input and output information

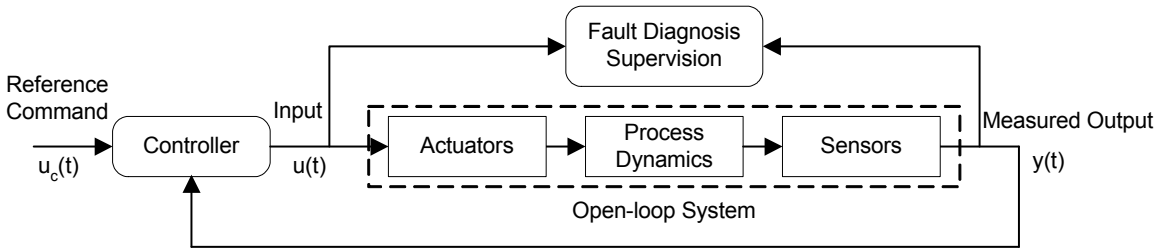

Figure 2.6: Fault Diagnosis and Control Loop

required in model-based FDI is related to the open-loop system. The performance of the controller would affect the performance of fault diagnosis, but the design of FDI scheme and the controller are not coupled (required simultaneously). This design methodology is incorporated in the thesis.

In the case when the input to the actuator $u(t)$ is not available, we have to use the reference command $u_c(t)$ in FDI, given that the control system is part of the system model. Therefore, the model involved is the relationship between the reference command $u_c(t)$ and the measured output $y(t)$, i.e., the closed-loop model. A robust controller may desensitize fault effects and make the diagnosis very difficult. This problem has been recognized by some researchers, e.g. [190]; [118], and the best solution is to design the fault diagnosis scheme and the controller simultaneously ([68]).

A state space model with the following continuous-discrete system form is used in the simulation studies developed in this thesis, and the process and measurement

equations are given by:

$$\begin{cases} \dot{x}(t) = f(x(t), u_R(t), t) + Q^{\frac{1}{2}} w(t) \\ z_k = h_k(x(t_k)) + R^{\frac{1}{2}} v_k \end{cases} \tag{2.20}$$

where

$x \in \Re^n$	state vector;
$z_k \in \Re^m$	output vector (measurable) at time t_k;
$u_R \in \Re^p$	actual input vector;
$w \in \Re^n$	modelling uncertainty vector, i.e., $N(0,1)$;
$Q \in \Re^n \times \Re^n$	process noise covariance matrix;
$v_k \in \Re^m$	measurement noise vector at time t_k, i.e., $N(0,1)$;
$R \in \Re^m \times \Re^m$	measurement noise covariance matrix;
f	nonlinear function of $x(t)$;
h_k	nonlinear function of at time t_k

Eq. 2.20 represents a problem where the process is continuous and the measurement is discrete. We assume that Q and R are known diagonal matrices. The process noise $Q^{\frac{1}{2}} w(t)$ can account for modelling uncertainties and unmeasurable disturbances to some extent.

For the purpose of fault modelling, an open-loop system can be separated into three parts: actuators, system dynamics and sensors as shown in Fig. 2.7. Accordingly, a fault can occur in any of these three parts, which then defines sensor, actuator or process faults.

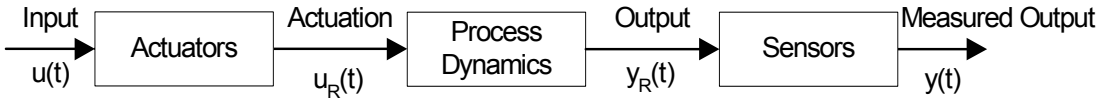

Figure 2.7: Open-Loop System

1. **Process Fault:**

When a process fault occurs in the system (see Fig. 2.8), the dynamic model of the system can be described as:

$$\dot{x}(t) = f_P(x(t), u_R(t), t) + Q^{\frac{1}{2}} w(t) \tag{2.21}$$

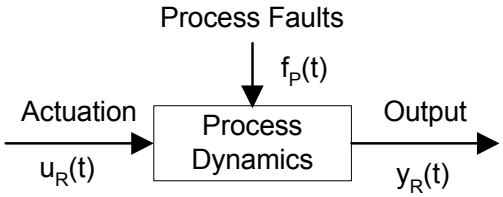

Figure 2.8: Process Fault

The process fault can be expressed as a change in the system parameters, for example a step change, ramp or random variations of the physical parameters. Without loss of generality, only additive faults are studied in this thesis. But the approach adopted here can always be extended to other forms of faults, i.e., multiplicative or other nonlinear functions, since the additive structure of the faults is not used explicitly in the FDI design. The analytical models of additive faults for a parameter c can have the following forms:

$$c(t) = c_0 + a \cdot U(t - t_f) \quad \text{step fault model} \tag{2.22}$$

$$c(t) = c_0 + \frac{a}{t_r} \cdot (t - t_f) \cdot [U(t - t_f) - U(t - t_f - t_r)] \quad \text{ramp fault model} \tag{2.23}$$

$$c(t) = c_0 + \left[\int_{t_f}^{t} v_q(\tau) d\tau \right]_{-a}^{a} \cdot U(t - t_f) \quad \text{random-walk fault model} \tag{2.24}$$

where

c_0	nominal value of the coefficient;
a	fault amplitude;
t_f	fault occurrence time;
t_r	fault rise time;
$U(t)$	unit step function;
$\left[\int_{t_f}^{t} v_q(\tau)d\tau\right]_{-a}^{a}$	random-walk process limited in amplitude in the range $[-a, a]$ and obtained by integration of the white noise process $v_q(t)$ with autocorrelation function $E\{v_q(t_1)v_q(t_2)\} = q\delta(t_1 - t_2)$.

Step, ramp and random-walk models can be used to simulate abrupt, slow and irregular variations of the physical parameters, respectively. Only step changes in the parameter are used to model fault occurrence in this thesis. Considering that the unit step function is an ideal description for mathematical convenience, a more practical step fault model with rise time is used in the simulation.

$$c(t) = \begin{cases} c_0 & t < t_f \\ c_0 + \frac{a}{t_r} \cdot (t - t_f) & t_f \leq t \leq t_f + t_r \\ c_0 + a & t > t_f + t_r \end{cases} \quad \text{step fault model with rise time} \tag{2.25}$$

2. **Sensor Fault:**

Generally speaking, the actual output of the system ($y_R(t)$) is not directly accessible, and sensors are then used to measure the system output.

Sensor faults are shown in Fig. 2.9 and can be described mathematically as an additive term (when the sensor dynamics are omitted):

$$y(t) = y_R(t) + f_s(t) \tag{2.26}$$

where $f_s \in \Re^m$ is the sensor fault vector. A step fault model with rise time is

used for sensor fault modelling.

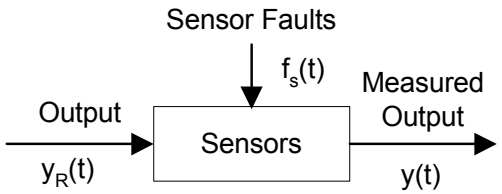

Figure 2.9: Sensor Fault

3. **Actuator Fault:**

It is also true that the actuator output ($u_R(t)$) of the system is often not directly accessible. For a controlled system, $u_R(t)$ is the actuator response to an actuator command $u(t)$ from the controller. This is shown in Fig. 2.10 and can be described as an additive term (when the actuator dynamics are omitted):

$$u_R(t) = u(t) + f_a(t) \tag{2.27}$$

where $f_a \in \Re^p$ is the actuator fault vector. Again, a step fault model with rise time is adopted for actuator fault modelling.

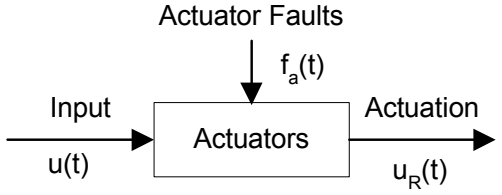

Figure 2.10: Actuator Fault

2.2 Multiple Model Extended Kalman Filter (MMEKF)

Kalman filtering ([70]; [24]) is an optimal state estimation process applied to a stochastic dynamic system. More precisely, for a linear stochastic system, the Kalman filter gives a linear, unbiased, and minimum error variance recursive algorithm to optimally estimate the unknown state of a dynamic system from noisy data taken at discrete time instants. The knowledge of the state (expressed by the covariance matrix) is improved at each step by taking the prior state estimates and new data for the subsequent state estimation.

A detailed derivation of the Kalman filter is given below.

2.2.1 Kalman Filter Preliminaries

2.2.1.1 Orthogonal Projection

Consider the linear stochastic system:

$$\begin{cases} x_{k+1} = A_k x_k + G_k w_k \\ y_k = C_k x_k + H_k v_k \end{cases} \tag{2.28}$$

where A_k, G_k, C_k, and H_k are known $n \times n$, $n \times p$, $m \times n$, and $m \times q$ matrices, respectively, with $1 \leq p, q, m \leq n$, and assume the basic random variables $\{x_0, w_0, w_1, v_0, v_1, \cdots\}$ are all independent and Gaussian with known statistics, i.e.,

$$\begin{aligned} &E(w_k) = 0, \quad E(w_k w_l^T) = Q_k \delta_{kl} \\ &E(v_k) = 0, \quad E(v_k v_l^T) = R_k \delta_{kl} \\ &E(w_k v_l^T) = 0, \; E(x_0 w_k^T) = 0, \quad E(x_0 v_k^T) = 0 \end{aligned} \tag{2.29}$$

for all $k, l = 0, 1, \cdots$, with Q_k and R_k being positive definite and symmetric matrices.

Let x be a random n-vector and w a random p-vector. We define the "inner product" $\langle x, w \rangle$ to be the $n \times p$ matrix

$$\langle x, w \rangle = cov(x, w) = E((x - E(x))(w - E(w))^T)$$

Let $\|w\|_p$ be the positive square root of $\langle w, w \rangle$, i.e., $\|w\|_p$ is a positive definite $p \times p$ matrix with

$$\|w\|_p^2 = \|w\|_p \|w\|_p^T = \langle w, w \rangle$$

Similarly, let $\|x\|_n$ be the positive square root of $\langle x, x \rangle$. Now, let $w_0, \cdots, w_r$ be a sequence of random p-vectors and consider the "linear span":

$$Z(w_0, \cdots, w_r) = \{z : z = \sum_{i=0}^{r} P_i w_i, P_0, \cdots, P_r, \ n \times p \text{ constant matrices}\}$$

First we look at a minimization problem to determine a $\hat{z}$ in $Z(w_0, \cdots, w_r)$ such that $tr \|x_k - \hat{z}\|_n^2 = F_k$, where

$$F_k := \min\{tr \|x_k - \hat{z}\|_n^2 : z \in Z(w_0, \cdots, w_r)\} \tag{2.30}$$

The following result characterizes $\hat{z}$.

Lemma 1 *[24] $\hat{z} \in Z(w_0, \cdots, w_r)$ satisfies $tr \|x_k - \hat{z}\|_n^2 = F_k$ if and only iff $\langle x_k - \hat{z}, w_j \rangle = 0_{n \times p}$ for all $j = 0, 1, \cdots, r$. Furthermore, $\hat{z}$ is unique in the sense that $tr \|x_k - \hat{z}\|_n^2 = tr \|x_k - \tilde{z}\|_n^2$ only if $\hat{z} = \tilde{z}$.*

Proof. Suppose that $tr \|x_k - \hat{z}\|_n^2 = F_k$ but $\langle x_k - \hat{z}, w_{j0} \rangle = C \neq 0_{n \times p}$ for some j_0 where $0 \leq j_0 \leq r$. Then $w_{j0} \neq 0$ so that $\|w_{j0}\|_p^2$ is a positive definite symmetric

matrix and so is its inverse $\|w_{j0}\|_p^{-2}$. Hence, $C\,\|w_{j0}\|_p^{-2}\,C^T \neq 0_{n\times n}$ and is a non-negative definite and symmetric matrix. It can be shown that

$$tr\{C\,\|w_{j0}\|_p^{-2}\,C^T\} > 0 \tag{2.31}$$

Now, the vector $\hat{z} + C\,\|w_{j0}\|_p^{-2}\,w_{j0}$ is in $Z(w_0, \cdots, w_r)$ and

$$\begin{aligned}
& tr\left\|x_k - (\hat{z} + C\,\|w_{j0}\|_p^{-2}\,w_{j0})\right\|_n^2 \\
= \;& tr\{\|x_k - \hat{z}\|_n^2 - \langle x_k - \hat{z}, w_{j0}\rangle\,(C\,\|w_{j0}\|_p^{-2})^T - C\,\|w_{j0}\|_p^{-2}\,\langle w_{j0}, x_k - \hat{z}\rangle \\
& + C\,\|w_{j0}\|_p^{-2}\,\|w_{j0}\|_p^2\,(C\,\|w_{j0}\|_p^{-2})^T\} \\
= \;& tr\{\|x_k - \hat{z}\|_n^2 - C\,\|w_{j0}\|_p^{-2}\,C^T\} \\
< \;& tr\,\|x_k - \hat{z}\|_n^2 = F_k
\end{aligned}$$

where Eq. 2.31 is used to obtain the last inequality. This contradicts the definition of F_k in Eq. 2.30.

Conversely, let $\langle x_k - \hat{z}, w_j\rangle = 0_{n\times p}$ for all $j = 0, 1, \cdots, r$. Let z be an arbitrary random n-vector in $Z(w_0, \cdots, w_r)$ and write $z_0 = z - \hat{z} = \sum_{j=0}^{r} P_{0j} w_j$ where P_{0j} are constant $n \times p$ matrices, $j = 0, 1, \cdots, r$. Then

$$\begin{aligned}
& tr\,\|x_k - z\|_n^2 \\
= \;& tr\,\|(x_k - \hat{z}) - z_0\|_n^2 \\
= \;& tr\{\|x_k - \hat{z}\|_n^2 - \langle x_k - \hat{z}, z_0\rangle - \langle z_0, x_k - \hat{z}\rangle + \|z_0\|_n^2\} \\
= \;& tr\{\|x_k - \hat{z}\|_n^2 - \sum_{j=0}^{r} \langle x_k - \hat{z}, w_j\rangle\,P_{0j}^T - \sum_{j=0}^{r} P_{0j}\,\langle x_k - \hat{z}, w_j\rangle^T + \|z_0\|_n^2\} \\
= \;& tr\,\|x_k - \hat{z}\|_n^2 + tr\,\|z_0\|_n^2 \\
\geq \;& tr\,\|x_k - \hat{z}\|_n^2
\end{aligned}$$

so that $tr\,\|x_k - \hat{z}\|_n^2 = F_k$. Furthermore, equality is attained if and only if $tr\,\|z_0\|_n^2 = 0$ or $z_0 = 0$ so that $z = \hat{z}$. This completes the proof of the lemma. ■

2.2.1.2 Innovation Sequences

The innovation sequences associated with a given random sequence is defined as follows:

Definition 2 *[24] Given a random m-vector sequence* $\{y_j\}$, $j = 0, 1, \cdots, k$. *The innovation sequence* $\{s_j\}$, $j = 0, 1, \cdots, k$, *of* $\{y_j\}$ *is defined by*

$$s_j = y_j - C_j \hat{z}_{j-1}, \;\; j = 0, 1, \cdots, k, \tag{2.32}$$

with $\hat{z}_{-1} = 0$ *and*

$$\hat{z}_{j-1} = \sum_{i=0}^{j-1} \hat{P}_{j-1,i} y_i \in Z(y_0, \cdots, y_{j-1}), \;\; j = 1, \cdots, k, \tag{2.33}$$

where the $m \times n$ *matrices* C_j *are the observation matrices in Eq. 2.28 and the* $n \times m$ *matrices* $\hat{P}_{j-1,i}$ *are chosen so that* $\hat{z}_{j-1}$ *solves the minimization problem of Eq. 2.30 with* $Z(w_0, \cdots, w_r)$ *replaced by* $Z(y_0, \cdots, y_{j-1})$.

The correlation property of the innovations sequence is given first.

Lemma 3 *[24] The innovation sequence* $\{s_j\}$ *of* $\{y_j\}$ *satisfies the following property:*

$$\langle s_j, s_l \rangle = (R_l + C_l \,\|x_l - \hat{z}_{l-1}\|_n^2 \, C_l^T)\delta_{jl},$$

where $R_l = Var(v_l) > 0$.

Proof. *For convenience, we set*

$$\hat{e}_j = C_j(x_j - \hat{z}_{j-1}) \tag{2.34}$$

We first observe that

$$s_j = \hat{e}_j + v_j \tag{2.35}$$

where $\{v_k\}$ *is the observation noise sequence, and*

$$\langle v_l, \hat{e}_j \rangle = 0_{m\times m} \quad \textit{for all } l \geq j \tag{2.36}$$

Clearly, Eq. 2.35 follows from Eq. 2.32, 2.34, and the observation equation in 2.28. Now, for $j = l$*, we have, by Eq. 2.35, 2.36, and 2.34 consecutively,*

$$\begin{aligned}
\langle s_l, s_l \rangle &= \langle \hat{e}_l + v_l, \hat{e}_l + v_l \rangle \\
&= \langle \hat{e}_l, \hat{e}_l \rangle + \langle v_l, v_l \rangle \\
&= C_l \left\| x_l - \hat{z}_{l-1} \right\|_n^2 C_l^T + R_l
\end{aligned}$$

For $j \neq l$*, since* $\langle \hat{e}_l, \hat{e}_j \rangle^T = \langle \hat{e}_j, \hat{e}_l \rangle$*, we can assume without loss of generality that* $j > l$*. Hence, by Eq. 2.35, 2.36, and lemma 1 we have*

$$\begin{aligned}
\langle s_j, s_l \rangle &= \langle \hat{e}_j, \hat{e}_l \rangle + \langle \hat{e}_j, v_l \rangle + \langle v_j, \hat{e}_l \rangle + \langle v_j, v_l \rangle \\
&= \langle \hat{e}_j, \hat{e}_l + v_l \rangle \\
&= \langle \hat{e}_j, s_l \rangle \\
&= \langle \hat{e}_j, y_l - C_l \hat{z}_{l-1} \rangle \\
&= \left\langle C_j(x_j - \hat{z}_{j-1}), y_l - C_l \sum_{i=0}^{l-1} \hat{P}_{l-1,i} y_i \right\rangle \\
&= C_j \langle x_j - \hat{z}_{j-1}, y_l \rangle - C_j \sum_{i=0}^{l-1} \langle x_j - \hat{z}_{j-1}, y_i \rangle \hat{P}_{l-1,i}^T C_l^T \\
&= 0_{m\times m}
\end{aligned}$$

This completes the proof of the lemma. ■

Since $R_j > 0$, lemma 3 says that $\{s_j\}$ is an "orthogonal" sequence of nonzero vectors which we can normalize by setting

$$e_j = \|s_j\|_m^{-1} s_j \tag{2.37}$$

Then $\{e_j\}$ is an "orthonormal" sequence in the sense that $\langle \hat{e}_i, \hat{e}_j \rangle = \delta_{ij} I_m$ for all i and j. Furthermore, it should be clear that

$$Z(e_0, \cdots, e_k) = Z(y_0, \cdots, y_k) \tag{2.38}$$

2.2.1.3 Minimum Variance Estimates

The minimum variance estimate $\hat{x}_k$ of the state vector x_k can be given by introducing the "Fourier expansion"

$$\hat{x}_k = \sum_{i=0}^{k} \langle x_k, e_i \rangle e_i \tag{2.39}$$

of x_k w.r.t the "orthonormal" sequence $\{e_j\}$. Since

$$\langle \hat{x}_k, e_j \rangle = \sum_{i=0}^{k} \langle x_k, e_i \rangle \langle e_i, e_j \rangle = \langle x_k, e_j \rangle ,$$

we have

$$\langle x_k - \hat{x}_k, e_j \rangle = 0_{n \times m}, \quad j = 0, 1, \cdots, k \tag{2.40}$$

It is also easy to see that

$$\langle x_k - \hat{x}_k, y_j \rangle = 0_{n \times m}, \quad j = 0, 1, \cdots, k \tag{2.41}$$

so that by lemma 1,

$$tr\,\|x_k - \hat{x}_k\|_n^2 = \min\{tr\,\|x_k - z\|_n^2 : z \in Z(y_0, \cdots, y_k)\}$$

That is, $\hat{x}_k$ is a minimum variance estimate of x_k.

2.2.1.4 Kalman Filtering Equations

From assumption 2.29, we first observe that

$$\langle w_{k-1}, e_i \rangle = 0_{p\times m}, \quad i = 0, 1, \cdots, k-1$$

so that

$$\begin{aligned}
\hat{x}_k &= \sum_{i=0}^{k} \langle x_k, e_i \rangle\, e_i \\
&= \sum_{i=0}^{k-1} \langle x_k, e_i \rangle\, e_i + \langle x_k, e_k \rangle\, e_k \\
&= \sum_{i=0}^{k-1} \{\langle A_{k-1}x_{k-1}, e_i \rangle\, e_i + \langle G_{k-1}w_{k-1}, e_i \rangle\, e_i\} + \langle x_k, e_k \rangle\, e_k \\
&= A_{k-1} \sum_{i=0}^{k-1} \langle x_{k-1}, e_i \rangle\, e_i + \langle x_k, e_k \rangle\, e_k \\
&= A_{k-1}\hat{x}_{k-1} + \langle x_k, e_k \rangle\, e_k
\end{aligned}$$

Hence, by defining

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1} \tag{2.42}$$

where $\hat{x}_{k-1} := \hat{x}_{k-1|k-1}$, we have

$$\hat{x}_k = \hat{x}_{k|k} = \hat{x}_{k|k-1} + \langle x_k, e_k \rangle\, e_k \tag{2.43}$$

We consider the random vector $(y_k - C_k\hat{x}_{k|k-1})$ and obtain the following:

Lemma 4 *[24] For $i = 0, 1, \cdots, k$,*

$$\left\langle y_k - C_k \hat{x}_{k|k-1}, e_i \right\rangle = \|s_k\|_m \, \delta_{ki}$$

Proof. *First we observe that*

$$\langle \hat{z}_i, s_k \rangle = 0_{n \times m}, \quad i = 0, 1, \cdots, k-1 \tag{2.44}$$

Hence, using Eq. 2.43, 2.40, and 2.44, we have

$$\begin{aligned}
& \left\langle y_k - C_k \hat{x}_{k|k-1}, e_k \right\rangle \\
= \; & \left\langle y_k - C_k(\hat{x}_{k|k} - \langle x_k, e_k \rangle \, e_k), e_k \right\rangle \\
= \; & \langle y_k, e_k \rangle - C_k \{ \left\langle \hat{x}_{k|k}, e_k \right\rangle - \langle x_k, e_k \rangle \} \\
= \; & \langle y_k, e_k \rangle - C_k \left\langle \hat{x}_{k|k} - x_k, e_k \right\rangle \\
= \; & \langle y_k, e_k \rangle \\
= \; & \left\langle s_k + C_k \hat{z}_{k-1}, \|s_k\|_m^{-1} \, s_k \right\rangle \\
= \; & \langle s_k, s_k \rangle \, \|s_k\|_m^{-1} + C_k \, \langle \hat{z}_{k-1}, s_k \rangle \, \|s_k\|_m^{-1} \\
= \; & \|s_k\|_m
\end{aligned}$$

On the other hand, using Eq. 2.43, 2.40, and 2.36, we have

$$\begin{aligned}
& \left\langle y_k - C_k \hat{x}_{k|k-1}, e_i \right\rangle \\
= \; & \left\langle C_k x_k + H_k v_k - C_k(\hat{x}_{k|k} - \langle x_k, e_k \rangle \, e_k), e_i \right\rangle \\
= \; & C_k \left\langle x_k - \hat{x}_{k|k}, e_i \right\rangle + H_k \, \langle v_k, e_i \rangle + C_k \, \langle x_k, e_k \rangle \, \langle e_k, e_i \rangle \\
= \; & 0_{m \times m}
\end{aligned}$$

for $i = 0, 1, \cdots, k-1$. This completes the proof. ■

It is clear, by using Eq. 2.38 and Eq. 2.39, that the random m-vector $(y_k - C_k\hat{x}_{k|k-1})$ can be expressed as $\sum_{j=0}^{k} M_j e_j$ for some constant $m \times m$ matrices M_j. It follows from lemma 4 that for $i = 0, 1, \cdots, k$,

$$\left\langle \sum_{j=0}^{k} M_j e_j, e_i \right\rangle = \|s_k\|_m \delta_{ki}$$

so that $M_0 = M_1 = \cdots = M_{k-1} = 0$ and $M_k = \|s_k\|_m$. Hence,

$$y_k - C_k\hat{x}_{k|k-1} = M_k e_k = \|s_k\|_m e_k$$

Define

$$K_k = \langle x_k, e_k \rangle \|s_k\|_m^{-1},$$

then we obtain

$$\langle x_k, e_k \rangle e_k = K_k(y_k - C_k\hat{x}_{k|k-1})$$

This, together with Eq. 2.43, gives the "prediction-correction" equation:

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}) \tag{2.45}$$

Remark 5 *$\hat{x}_{k|k}$ is an unbiased estimate of x_k by choosing an appropriate initial estimate. In fact,*

$$\begin{aligned} & x_k - \hat{x}_{k|k} \\ = \; & A_{k-1}x_{k-1} + G_{k-1}w_{k-1} - A_{k-1}\hat{x}_{k-1|k-1} - K_k(y_k - C_kA_{k-1}\hat{x}_{k-1|k-1}) \end{aligned}$$

By using $y_k = C_kx_k + H_kv_k = C_kA_{k-1}x_{k-1} + C_kG_{k-1}w_{k-1} + H_kv_k$, we have

$$\begin{aligned} & x_k - \hat{x}_{k|k} \\ = \; & (I - K_kC_k)A_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1}) + (I - K_kC_k)G_{k-1}w_{k-1} - K_kH_kv_k \end{aligned} \tag{2.46}$$

Since the noise sequences have zero-mean, we have

$$E(x_k - \hat{x}_{k|k}) = (I - K_k C_k) A_{k-1} E(x_{k-1} - \hat{x}_{k-1|k-1})$$

so that

$$E(x_k - \hat{x}_{k|k}) = (I - K_k C_k) A_{k-1} \cdots (I - K_1 C_1) A_0 E(x_0 - \hat{x}_{0|0}).$$

Hence, if we set

$$\hat{x}_{0|0} = E(x_0) \tag{2.47}$$

then $E(x_k - \hat{x}_{k|k}) = 0$ *or* $E(\hat{x}_{k|k}) = E(x_k)$ *for all k, i.e.,* $\hat{x}_{k|k}$ *is an unbiased estimate of* x_k.

Now we are going to derive a recursive formula for K_k. Using Eq. 2.41 and Eq. 2.46, we first have

$$\begin{aligned}
0 &= \langle x_k - \hat{x}_k, y_k \rangle \\
&= \big\langle (I - K_k C_k) A_{k-1} (x_{k-1} - \hat{x}_{k-1|k-1}) + (I - K_k C_k) G_{k-1} w_{k-1} - K_k H_k v_k, \\
&\quad C_k A_{k-1} ((x_{k-1} - \hat{x}_{k-1|k-1}) + \hat{x}_{k-1|k-1}) + C_k G_{k-1} w_{k-1} + H_k v_k \big\rangle \\
&= (I - K_k C_k) A_{k-1} \left\| x_{k-1} - \hat{x}_{k-1|k-1} \right\|_n^2 A_{k-1}^T C_k^T \\
&\quad + (I - K_k C_k) G_{k-1} Q_{k-1} G_{k-1}^T C_k^T - K_k H_k R_k H_k^T
\end{aligned} \tag{2.48}$$

where we have used the fact that $\left\langle x_{k-1} - \hat{x}_{k-1|k-1}, \hat{x}_{k-1|k-1} \right\rangle = 0_{n \times n}$, a consequence of lemma 1, and

$$\begin{aligned}
\langle x_k, w_k \rangle = 0_{n \times p}, \quad \left\langle \hat{x}_{k|k}, w_j \right\rangle = 0_{n \times p}, \\
\langle x_k, v_j \rangle = 0_{n \times m}, \quad \left\langle \hat{x}_{k-1|k-1}, v_k \right\rangle = 0_{n \times m},
\end{aligned} \tag{2.49}$$

$i = 0, 1, \cdots, k$. Define

$$P_{k|k} = \left\| x_k - \hat{x}_{k|k} \right\|_n^2$$

and

$$P_{k|k-1} = \left\| x_k - \hat{x}_{k|k-1} \right\|_n^2$$

Then again by Eq. 2.49 we have

$$\begin{aligned} P_{k|k-1} &= \left\| A_{k-1}x_{k-1} + G_{k-1}w_{k-1} - A_{k-1}\hat{x}_{k-1|k-1} \right\|_n^2 \\ &= A_{k-1} \left\| x_{k-1} - \hat{x}_{k-1|k-1} \right\|_n^2 A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \\ &= A_{k-1}P_{k-1|k-1}A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \end{aligned} \tag{2.50}$$

On the other hand, from Eq. 2.48, we also have

$$(I - K_kC_k)A_{k-1}P_{k-1|k-1}A_{k-1}^TC_k^T + (I - K_kC_k)G_{k-1}Q_{k-1}G_{k-1}^TC_k^T - K_kH_kR_kH_k^T = 0$$

In solving for K_k from this expression, we write

$$\begin{aligned} & K_k[H_kR_kH_k^T + C_k(A_{k-1}P_{k-1|k-1}A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T)C_k^T] \\ = \; & [A_{k-1}P_{k-1|k-1}A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T]C_k^T = P_{k|k-1}C_k^T \end{aligned}$$

and obtain

$$K_k = P_{k|k-1}C_k^T(H_kR_kH_k^T + C_kP_{k|k-1}C_k^T)^{-1} \tag{2.51}$$

where $H_kR_kH_k^T$ is positive definite and $C_kP_{k|k-1}C_k^T$ is non-negative definite so that the inverse of their sum always exists.

Next, we will obtain a recursive scheme for $P_{k|k}$ in terms of $P_{k|k-1}$.

$$\begin{aligned}
P_{k|k} &= \left\|x_k - \hat{x}_{k|k}\right\|_n^2 \\
&= \left\|x_k - (\hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}))\right\|_n^2 \\
&= \left\|x_k - \hat{x}_{k|k-1} - K_k(C_k x_k + H_k v_k) + K_k C_k \hat{x}_{k|k-1}))\right\|_n^2 \\
&= \left\|(I - K_k C_k)(x_k - \hat{x}_{k|k-1}) - K_k H_k v_k\right\|_n^2 \\
&= (I - K_k C_k) P_{k|k-1} (I - K_k C_k)^T + K_k H_k R_k H_k^T K_k^T
\end{aligned}$$

where we used the result $\left\langle x_k - \hat{x}_{k|k-1}, v_k \right\rangle = 0_{n\times m}$. This relation can be further simplified by using Eq. 2.51 since

$$\begin{aligned}
& (I - K_k C_k) P_{k|k-1} (K_k C_k)^T \\
= \;& P_{k|k-1} C_k^T K_k^T - K_k C_k P_{k|k-1} C_k^T K_k^T \\
= \;& K_k H_k R_k H_k^T K_k^T,
\end{aligned}$$

we have

$$\begin{aligned}
P_{k|k} &= (I - K_k C_k) P_{k|k-1} (I - K_k C_k)^T + (I - K_k C_k) P_{k|k-1} (K_k C_k)^T \\
&= (I - K_k C_k) P_{k|k-1} \qquad (2.52)
\end{aligned}$$

Therefore, combining Eqs. 2.42, 2.45, 2.47, 2.50, 2.51, and 2.52, together with

$$P_{0|0} = \left\|x_0 - \hat{x}_{0|0}\right\|_n^2 = Var(x_0), \qquad (2.53)$$

we obtain the Kalman filtering equations as follows:

$$\begin{cases} P_{0|0} = Var(x_0) \\ P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \\ K_k = P_{k|k-1}C_k^T(H_kR_kH_k^T + C_kP_{k|k-1}C_k^T)^{-1} \\ P_{k|k} = (I - K_kC_k)P_{k|k-1} \\ \hat{x}_{0|0} = E(x_0) \\ \hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1} \\ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}) \end{cases} \tag{2.54}$$

where $k = 1, 2, \cdots$.

2.2.2 Extended Kalman Filter Equations

The Kalman filtering process is designed to estimate the state vector in a linear model. For a nonlinear system, a nonlinear filter is needed. The exact solution to a general nonlinear filtering problem involves the existence of solutions to partial differential equations, is generally too complicated for practical implementation, and the existence of a finite dimensional filter realization is generally unknown. Some research attempts have been made in the nonlinear observer design ([44]; [146]; [154]). However, they are only applicable to special classes of nonlinear systems. The most common approach is to use an Extended Kalman Filter (EKF), which uses nonlinear state equations, but linearizes the process and measurement nonlinearity around the latest state estimates for covariance propagation.

Consider a nonlinear model of the form

$$\begin{cases} x_{k+1} = f_k(x_k) + \Gamma_k(x_k) w_k \\ y_k = h_k(x_k) + H_k v_k \end{cases} \tag{2.55}$$

where f_k and h_k are vector-valued functions in $\Re^n$ and $\Re^m$, respectively, $1 \leq m \leq n$ and Γ_k is a matrix-valued function in $\Re^{n \times m}$, such that for each k the first order partial derivatives of $f_k(x_k)$ and $h_k(x_k)$ w.r.t. all the components of x_k are continuous. As before, we simply consider zero-mean Gaussian white noise sequences $\{w_k\}$ and $\{v_k\}$ in $\Re^p$ and $\Re^q$, respectively, $1 \leq p, q \leq n$, and

$$E(w_k w_l^T) = Q_k \delta_{kl}, \quad E(v_k v_l^T) = R_k \delta_{kl}$$

$$E(w_k v_l^T) = 0, \; E(x_0 w_k^T) = 0, \quad E(x_0 v_k^T) = 0$$

for all k and l. The extended Kalman filter equations are analogous to the normal Kalman filter equations.

In order to be consistent with the linear model, the initial estimate $\hat{x}_0 = \hat{x}_{0|0}$ and the predicted state $\hat{x}_{1|0}$ are chosen to be

$$\hat{x}_0 = E(x_0), \quad \hat{x}_{1|0} = f_0(\hat{x}_0)$$

Then define $\hat{x}_k = \hat{x}_{k|k}$, consecutively, for $k = 1, 2, \cdots$, using the predicted state equation

$$\hat{x}_{k+1|k} = f_k(\hat{x}_k) \tag{2.56}$$

and the linear state-space description

$$\begin{cases} x_{k+1} = A_k x_k + u_k + G_k w_k \\ z_k = C_k x_k + H_k v_k \end{cases} \tag{2.57}$$

where A_k, u_k, G_k, z_k, and C_k are to be determined in real-time. Suppose that $\hat{x}_i$ has been determined and $\hat{x}_{i+1|i}$ is defined using Eq. 2.56, for $i = 1, 2, \cdots, k$. We consider the first order Taylor approximation of $f_k(x_k)$ at $\hat{x}_k$ and that of $h_k(x_k)$ at $\hat{x}_{k|k-1}$, that is,

$$\begin{cases} f_k(x_k) \simeq f_k(\hat{x}_k) + A_k(x_k - \hat{x}_k) \\ h_k(x_k) \simeq h_k(\hat{x}_{k|k-1}) + C_k(x_k - \hat{x}_{k|k-1}) \end{cases} \tag{2.58}$$

where

$$A_k = \left[\frac{\partial f_k}{\partial x_k}(\hat{x}_k)\right] \text{ and } C_k = \left[\frac{\partial h_k}{\partial x_k}(\hat{x}_{k|k-1})\right] \tag{2.59}$$

Here and throughout, for any vector-valued function

$$g(x_k) = \begin{bmatrix} g_1(x_k) \\ \vdots \\ g_m(x_k) \end{bmatrix}$$

where

$$x_k = \begin{bmatrix} x_k^1 \\ \vdots \\ x_k^n \end{bmatrix}$$

we denote

$$\left[\frac{\partial g}{\partial x_k}(x_k^*)\right] = \begin{bmatrix} \frac{\partial g_1}{\partial x_k^1}(x_k^*) & \cdots & \frac{\partial g_1}{\partial x_k^n}(x_k^*) \\ \vdots & & \vdots \\ \frac{\partial g_m}{\partial x_k^1}(x_k^*) & \cdots & \frac{\partial g_m}{\partial x_k^n}(x_k^*) \end{bmatrix}$$

Hence, by setting

$$\begin{cases} u_k = f_k(\hat{x}_k) - A_k\hat{x}_k \\ G_k = \Gamma_k(\hat{x}_k) \\ z_k = y_k - h_k(\hat{x}_{k|k-1}) + C_k\hat{x}_{k|k-1} \end{cases} \tag{2.60}$$

the nonlinear model (Eq. 2.55) is approximated by the linear model (Eq. 2.57) using the matrices and vectors defined in Eqs. 2.59 and 2.60 at the k^{th} instant. Of course, this linearization is possible only if $\hat{x}_k$ has been determined. First we already know the initial estimate $\hat{x}_0$, so that the system equation in Eq. 2.57 is valid for $k = 0$. From this, we define $\hat{x}_1 = \hat{x}_{1|1}$ as the optimal unbiased estimate (with optimal weight) of x_1 in the linear model (Eq. 2.57), using the data $[y_0^T \;\; z_1^T]^T$. Now, by applying Eq. 2.56, Eq. 2.57 is established for $k = 1$, so that $\hat{x}_2 = \hat{x}_{2|2}$ can be determined similarly, using the data $[y_0^T \;\; z_1^T \;\; z_2^T]^T$, etc. From the Kalman filter results for a linear system (Eq. 2.54), we arrive at the "correction" formula

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - C_k\hat{x}_{k|k-1}) \\ &= \hat{x}_{k|k-1} + K_k(y_k - h_k(\hat{x}_{k|k-1})) \end{aligned}$$

where K_k is the Kalman filter gain matrix for the linearized model (Eq. 2.57) at the kth instant.

The *Extended Kalman Filter* block diagram is shown in Fig. 2.11. The resulting EKF algorithm is summarized as follows:

$$\begin{cases} P_{0|0} = Var(x_0), \quad \hat{x}_0 = E(x_0) \;\; \text{for } k = 1, 2, \cdots, \\ P_{k|k-1} = \left[\frac{\partial f_{k-1}}{\partial x_{k-1}}(\hat{x}_{k-1})\right] P_{k-1|k-1} \left[\frac{\partial f_{k-1}}{\partial x_{k-1}}(\hat{x}_{k-1})\right]^T + \Gamma_{k-1}(\hat{x}_{k-1}) Q_{k-1} \Gamma_{k-1}(\hat{x}_{k-1})^T \\ \hat{x}_{k|k-1} = f_{k-1}(\hat{x}_{k-1}) \\ K_k = P_{k|k-1} \left[\frac{\partial h_k}{\partial x_k}(\hat{x}_{k|k-1})\right]^T (H_k R_k H_k^T + \left[\frac{\partial h_k}{\partial x_k}(\hat{x}_{k|k-1})\right] P_{k|k-1} \left[\frac{\partial h_k}{\partial x_k}(\hat{x}_{k|k-1})\right]^T)^{-1} \\ P_{k|k} = (I - K_k \left[\frac{\partial h_k}{\partial x_k}(\hat{x}_{k|k-1})\right]) P_{k|k-1} \\ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - h_k(\hat{x}_{k|k-1})) \end{cases} \tag{2.61}$$

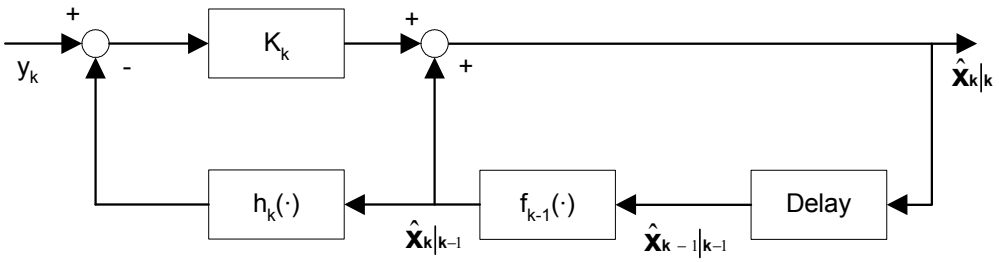

Figure 2.11: EKF Block Diagram

2.2.3 Multiple Model EKF FDI System

The idea of using the Kalman filter as a FDI tool is based on a well known fact for a linear stochastic system (Eq. 2.28): the residual (innovation sequence), which is the difference between the measurement and the predicted measurement from the Kalman filter, is a zero-mean white sequence and is unbiased when the Kalman filter matches the system model.

Remark 6 *This fact applies whether the system is fault-free or under a fault condition, as long as the correct model is being used. The standard approach of using the Kalman filter as a FDI tool is to design a Kalman filter under normal condition, fault detection can then be achieved by monitoring the zero-mean whiteness properties of the residual. But if one can design the Kalman filter under a particular fault condition, the residual when the system is under that particular fault condition should again be zero-mean and white. The violation of the zero-mean white condition not only indicates a fault occurs, but also shows which fault occurs by determining the residual of the particular Kalman filter model that satisfies the zero-mean white hypothesis.*

Hence, the motivation of the multiple model Kalman filter approach to FDI is to design a bank of Kalman filters, where each filter is tuned to a specific fault. Using its embedded model, each filter predicts values for the measurement variables. The residual is an indicator of how well the filter is performing, i.e. matches the operating characteristics of the plant.

Using a bank of Kalman filters was pioneered by Magill ([96]) who used a parallel structure of estimators in order to estimate a sampled stochastic process. For other typical applications, please refer to Multiple Model Filters (MMFs) in section 2.1.3.

The plant model used in the simulation study is a highly nonlinear chemical process. Therefore, the EKF is chosen to estimate the unmeasurable state variables and the entire FDI system is based on a bank of EKFs.

The MMEKF system (see Fig. 2.12) consists of a bank of parallel extended Kalman filters, each with a different internal model. The Kalman filters are provided a measurement vector (y) and the input vector (u), and produce a state estimate ($\hat{x}_k$, k is the time instant) and a residual (r_i, $i = 0, 1, \cdots, n$). The normal EKF model provides the best estimate for the system under nominal operating conditions. Other EKF models incorporate the fault effects in their structures and represent different fault types. Each of these filters is built with an embedded fault model, and an increase in robustness is expected since each filter uses additional knowledge about the expected fault. Since each Kalman filter is sensitive to a particular fault, a small residual would suggest a match between the actual measurement and the corresponding filter, and therefore a possible fault, so the size of the residuals from various

filters gives a relative indication of how adequately each of these models represents the actual fault status. Therefore, this multiple model approach has an inherent fault isolation capability. A residual evaluation mechanism is used to generate the fault index, which assists in the fault decision making process.

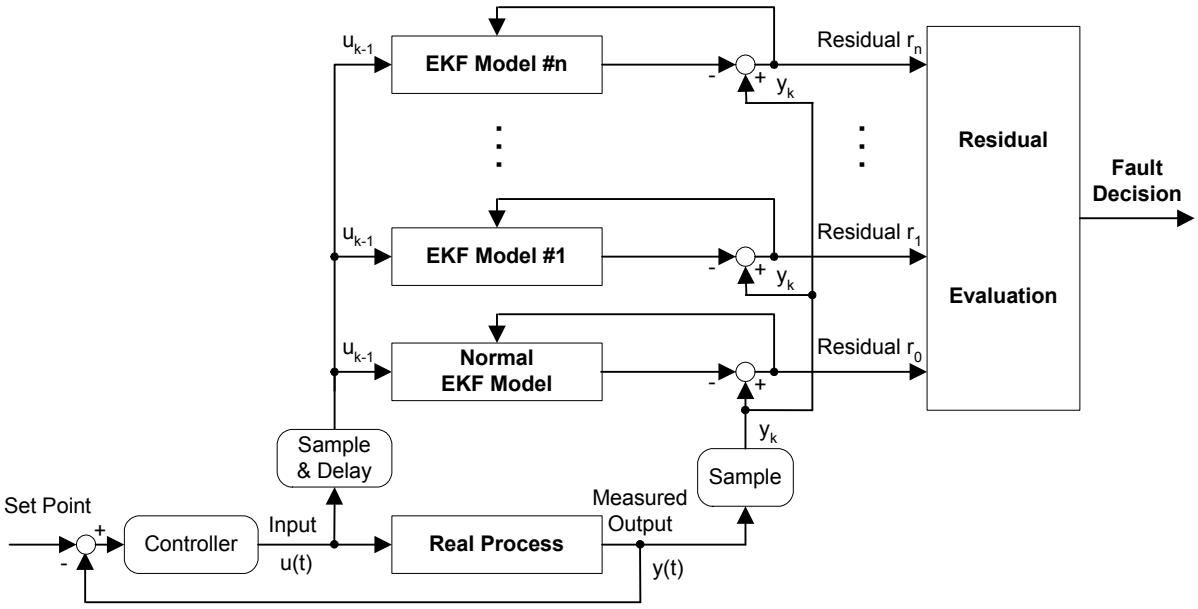

Figure 2.12: MMEKF FDI System Structure

A schematic diagram of a single EKF and its relationship to the real process is shown in Fig. 2.13. It is easy to see that the EKF is in parallel with the real system, and has almost the identical structure. Inputs to the EKF system include the controller information u and the measurement information y from the sensor (via residuals which function as feedback signals to the EKF system). This conforms with the real situation since these two types of input information are the only available information from the real process. The rule of thumb in designing the EKF system is trying to match the EKF model with the real situation as close as possible. In other

words, with knowledge of the controller information and the measurement information, the EKF model should be able to predict the state variables of the real process, no matter if it is operating under the normal condition or a fault (including actuator, sensor, and process fault) condition.

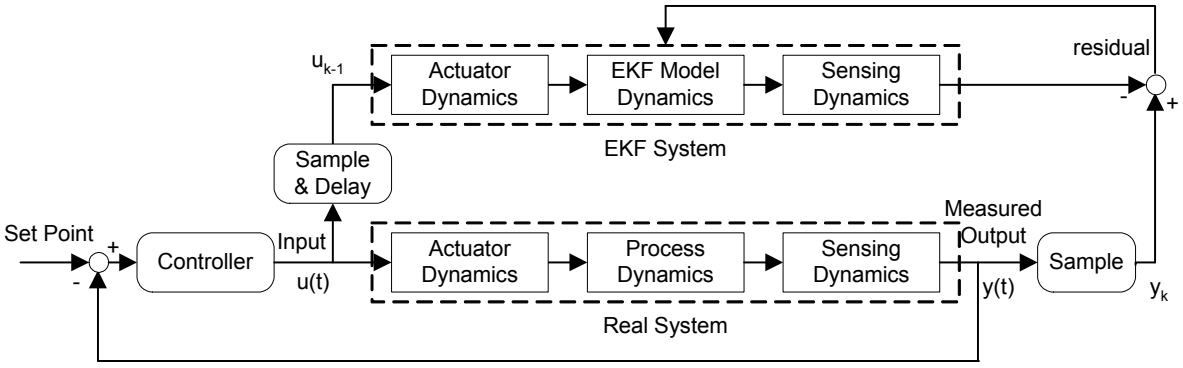

Figure 2.13: EKF vs Real System Schematic Diagram

Remark 7 *Faults in this thesis study have been classified into three categories: actuator, sensor, and process fault. To model the EKF system under various faults correctly, understanding the type and location of the fault in the system is important. Detailed examples of fault modelling will be shown in the simulation study sec. 2.4.1.3.*

2.3 Residual Evaluation Using Principal Component Analysis

2.3.1 Principal Component Analysis (PCA) Preliminaries

2.3.1.1 Introduction

Principal Component Analysis (PCA) is defined as a linear transformation of the original correlated variables into a new set of variables that are uncorrelated with each other. The original variables are assumed independent, each with a multivariate normal distribution. The linear transformation is:

$$T = X \cdot P \tag{2.62}$$

where $X \in \Re^{n \times p}$, a matrix of n observations with p variables, measured about their means, the transformation matrix $P \in \Re^{p \times p}$ is called the loading matrix, which is the matrix of eigenvectors of the sample covariance or correlation matrix of the original data, and T is called score matrix, which is the projection of the original data into the principal component subspace. From this point on, principal component is abbreviated as PC.

The main attribute of PCA is that it enables the dimensionality of the problem to be reduced. If some of the variables in the data set are collinear or highly correlated, then a smaller number of principal components q are required to explain the majority of the variance in the data. In practice, the variance left unexplained by the PCs is captured by the residual subspace, which are often associated with the instrument or

process noise.

A PCA model can be established based on the historical data collected when only the common causes of process variation, which means the process is at steady state, are present. Future observations can be referenced against this "in-control" model. A new multivariate observation x_{new} can be projected onto the principal component subspace defined by the PCA loading matrix to obtain the score vector ($t_{new} = P_q^T x_{new}$), and onto the residual subspace to obtain the residual ($e_{new} = x_{new} - \hat{x}_{new}$), where $\hat{x}_{new} = P_q t_{new}$, and P_q is the first q columns in the loading matrix that defines the PC projection. A detailed geometric interpretation of PCA is given in section 2.3.1.3.

Two important statistical measures, the T^2 score and the SPE (squared prediction error, also referred to as the Q statistic), are used to monitor the process. The control limits of these two statistical measures are calculated based on the reference historical data. New observations are then checked against the control limits to detect any abnormal events. Instead of monitoring multiple variables at the same time, only two variables need to be monitored in the PCA statistical monitoring. The T^2 and Q statistics can serve as fault detection indicators. Once T^2 and/or Q scores of the new observation exceed the control limits, an alarm is triggered.

2.3.1.2 Singular Value Decomposition (SVD)

The kernel of the PCA technique is SVD (Singular Value Decomposition).

An arbitrary matrix $X \in \Re^{n \times p}$ of n observations on p variables, measured about

their means, can be decomposed into the form:

$$X = U\Sigma V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_p u_p v_p^T \tag{2.63}$$

where U contains the left singular vectors $(u_i, i = 1, 2, \cdots, p)$, V contains the right singular vectors $(v_i, i = 1, 2, \cdots, p)$ and $\sigma_i, i = 1, 2, \cdots, p$ are the singular values.

There are two decomposition forms:

1. $U_{(n\times p)} = [u_1\ u_2\ \cdots\ u_p]$, $V_{(p\times p)} = [v_1\ v_2\ \cdots\ v_p]$ are orthonormal matrices.

   $\Sigma_{(p\times p)} = diag[\sigma_1, \sigma_2, \cdots, \sigma_p]$ is a real, non-negative diagonal matrix. Its diagonal elements are the singular values $\sigma_i,\ i = 1, 2, \cdots, p$, which are the absolute value of the square roots of the eigenvalues of $X^T X$, ordered according to $\sigma_1 > \sigma_2 > \sigma_p \geq 0$.

   The columns of U, $u_i, i = 1, 2, \cdots, p$, are the normalized eigenvectors of XX^T.

   $$XX^T U = U\Sigma^2 = [u_1\ u_2 \cdots u_p] \cdot diag[\sigma_1^2, \sigma_2^2, \cdots, \sigma_p^2] = [\sigma_1^2 u_1\ \sigma_2^2 u_2 \cdots \sigma_p^2 u_p]$$

   The columns of V, $v_i, i = 1, 2, \cdots, p$, are the normalized eigenvectors of $X^T X$.

   $$X^T XV = V\Sigma^2 = [v_1\ v_2 \cdots v_p] \cdot diag[\sigma_1^2, \sigma_2^2, \cdots, \sigma_p^2] = [\sigma_1^2 v_1\ \sigma_2^2 v_2 \cdots \sigma_p^2 v_p]$$

2. $U_{(n\times n)} = [u_1\ u_2\ \cdots\ u_n]$, $V_{(p\times p)} = [v_1\ v_2\ \cdots\ v_p]$ are orthonormal matrices.

   $\Sigma_{(n\times p)}$ is defined as $\begin{cases} \Sigma_{(i\times j)} \geq 0,\ i = j \\ \Sigma_{(i\times j)} = 0,\ i \neq j \end{cases}$

   The number of non-zero values in Σ equals to the rank of X.

Remark 8 *PCA can be calculated by eigenvalue decomposition of the covariance matrix S or correlation matrix R of the data matrix X. When the variables have the same magnitude, the covariance matrix is used for PCA. Otherwise, the variables have to be scaled to zero mean and unit variance before calculating the correlation matrix. The eigenvalue decomposition of S or R is based on the SVD of the data matrix X. Note that PCA is a scale-dependent method. If correlation based PCA is used, it depends on the mean and standard deviation of the data matrix X.*

Remark 9 *SVD is a very important method for PCA. It provides a computationally efficient method of actually finding the PCs, i.e., the loading matrix P in Eq. 2.62 can be obtained from Eq. 2.63 if we let $P = V$. As a bonus we also get, in U, a standardized version of the score matrix if we let $T = U\Sigma$ (see Eq. 2.63) since $V^TV = I_p$.*

2.3.1.3 Geometric Interpretation of PCA

Now let's take a look at PCA in a geometric way. Consider the measurement data matrix $X \in \Re^{n\times p}$, let $P_q \in \Re^{p\times q}$ be the PCs of the loading matrix (first q columns of the loadings matrix P), q is the number of PCs selected. Any sample vector x $(p \times 1)$ in the measurement space can be decomposed into two parts, $\hat{x}$ and $\tilde{x}$, corresponding to the projection onto the PC subspace ℓ_p and the Residual subspace ℓ_r, respectively. Matrices $P_qP_q^T$ and $I - P_qP_q^T$ span the PC subspace and the residual subspace, i.e., $\hat{x} = P_qP_q^Tx$ and $\tilde{x} = (I - P_qP_q^T)x$, and these vectors are orthogonal, i.e. $(I - P_qP_q^T)P_qP_q^T = 0$.

Fig. 2.14 illustrates an example of the geometric interpretation of PCA in the 3-dimensional case. The sample vector x is decomposed into the orthogonal projections $\hat{x}$ and $\tilde{x}$ on the PC subspace (ℓ_p), which is represented by the green ellipse, and the residual subspace (ℓ_r), respectively. In this example, the PC subspace has dimension 2 and the residual subspace has dimension 1.

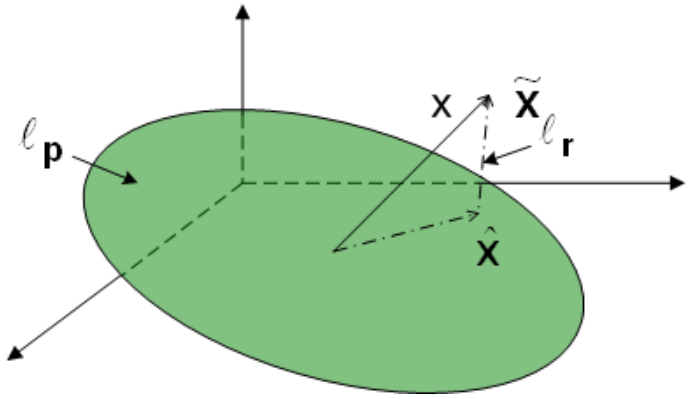

Figure 2.14: An Example of the Geometric Interpretation of PCA

2.3.1.4 T^2 and Q statistics

Two statistics, T^2 and Q, are used for multivariate process fault detection using the PCA method. Hotelling's T^2 statistic is given by

$$T^2 = \sum_{i=1}^{q} \frac{t_i^2}{s_{ti}^2} \tag{2.64}$$

where t_i, $(i = 1, 2, \ldots, p)$, is the i^{th} component of the score vector $t = P^T x$, s_{ti}^2 is the variance of t_i, which is also the i^{th} eigenvalue of the sample covariance matrix S and q is the number of PCs. The T^2 statistic measures the variation of the score vector in the PC space. It forms an ellipse which represents the joint limits of variations that

can be explained by the PCA model and this statistic can be used to detect whether or not the variation in the PC subspace is greater than the normal variability that can be explained by the PCA model.

T^2 statistic will only detect variation within the PC subspace that is greater than what can be explained by the PCA model. When a new type of event occurs that was not present in the PCA model, the new observation will move away from the PC subspace. This type of event can be detected by the Q statistic, which measures the variation in the residual subspace. The Q statistic (SPE) is defined as:

$$Q = e^T e = (x - \hat{x})^T (x - \hat{x}) \tag{2.65}$$

where $e = (I - P_q P_q^T)x$ is the residual of the sample x. It represents the squared distance of each observation perpendicular to the PC subspace and measures unstructured residuals that can not be accounted for by the PCA model.

The T^2 and Q statistics are complementary to each other, and jointly they can measure the variation in the entire measurement space.

Remark 10 *Out-of-control or abnormal points can be detected by comparing these two statistics with their control limits. Detection by either one of these two methods is sufficient to declare that a sample is possibly out-of-control.*

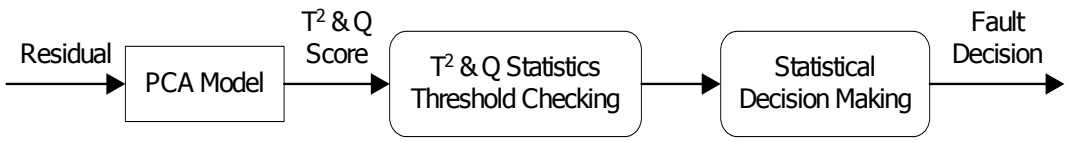

Figure 2.15: Residual Evaluation Process

2.3.2 PCA Residual Evaluation

2.3.2.1 PCA Fault Evaluation Mechanism

The PCA residual evaluation process is depicted in Fig. 2.15. One of the key issues here is the design of the PCA model. For PCA model under nominal condition, it is relatively easy. But for situation when a fault occurs, special care should be taken on the choice of the fault magnitude. The concern originates from the way we model the faulty condition, as you will see in the simulation study section. The fault is usually modelled as a deviation from the nominal value of certain parameter. The magnitude of this deviation is important for defining the faulty condition. In the simulation study, we define a fault magnitude range $[f_{min}\ f_{max}]$ for each fault type, where f_{min} is the minimum detectable fault magnitude when the EKF model is under normal condition and the real process is under the faulty condition, f_{max} is the largest possible fault magnitude in which the system remains stable. The determination of f_{min} is based on the control limits of the PCA model when both the EKF model and the real process are under the normal operating condition. Therefore, the fault magnitude range $[f_{min}\ f_{max}]$ defines the admissible fault range which is sufficient for delectability and small enough to keep the system stable.

The analysis in the extended Kalman filter section shows that the residual between the measurement and the predicted measurement from the Kalman filter, is a zero-mean white sequence when both the EKF model and the real process are under the same operating condition, normal or faulted operation with the same fault magnitude. While extensive simulations show that a slight mismatch between the fault magnitude of the EKF model and the real process would make the residual deviate from its zero-mean white behavior, and in turn the T^2 and Q scores would exceed their corresponding control limits. The sensitivity to the magnitude of the mismatch varies with fault type. This observation gives us an idea of using MMEKF as a tool for fault evaluation, trying to find the matching fault type and magnitude of the EKF model w.r.t. the real process which makes the T^2 and Q scores stay within their corresponding control limits. Now the fault evaluation problem becomes a search problem for the fault magnitude that minimizes the T^2 and Q scores. Fortunately the search interval is finite because of the admissible fault magnitude range $[\mathrm{f}_{\min}\ \mathrm{f}_{\max}]$, and the cost functions, which in this case are T^2 and Q scores, are unimodal, which is shown in the next section.

Remark 11 *Two fundamental statistical assumptions when applying PCA are: measured variables are independently identical and normally distributed; measured variables are uncorrelated in time (not auto-correlated). In practice, measurements from dynamic chemical processes do not satisfy these two assumptions, resulting in a loss of the statistical basis for the T^2 and Q statistics. However, residuals of MMEKF satisfy these two assumptions, which make PCA a perfect tool for residual evaluation*

using a MMEKF FDI structure.

One thing remains to be clarified is how to decide the reference PCA model for each fault type. The residual is different under different types of faults and fault magnitudes, the question is for each fault type which fault magnitude should be chosen to define the reference residual data set? Fortunately, simulation studies show that residuals under different fault magnitudes, when the EKF model matches the real process, are all zero-mean and their correlation structures are similar, which is illustrated in the next section. Therefore, the fault magnitude is chosen such that the reference PCA model represents the average correlation structure of the residuals.

We have two methods for setting up the reference PCA model. For each fault mode i, find the corresponding admissible fault magnitude range, $[f_{i,\min} f_{i,\max}]$. Use either $(f_{i,\max} + f_{i,\min})/2$ or the $(f_{i,\min}, f_{i,\max})$ pair to run the simulation and get the reference residual data set, where both the EKF model and the real process are under the same faulty condition with the same fault magnitude. If $(f_{i,\max}+f_{i,\min})/2$ is chosen to generate the reference residual data set, its correlation matrix is used to construct the PCA model. If the $(f_{i,\min}, f_{i,\max})$ pair is chosen to generate the reference residual data set, the lumped/pooled correlation matrix, $R = \frac{(n_1-1)R_1+(n_2-1)R_2}{n_1+n_2-2}$, is used to construct the PCA model, where n_1 is the number of observations in reference set $f_{i,\min}$ and n_2 is the number of observations in reference set $f_{i,\max}$, R_1 is the correlation matrix of reference set $f_{i,\min}$ and R_2 is the correlation matrix of reference set $f_{i,\max}$.The resulting PCA model should be able to represent the typical residual behavior of the corresponding fault condition. The first method is selected in the simulation study.

2.3.2.2 Implementation Issues

There are some key issues that need to be considered while developing the PCA model.

It is necessary to normalize the data before using the PCA. In our application, the data is normalized to zero mean, unit variance to eliminate the undesired weight difference in the variation of different variables with different magnitudes. This correlation PCA approach treats every variable equally and gives each variable an equal weight in the total variance.

1. **Selecting the Proper Number of Principal Components**

   A key issue in developing a PCA model is to choose an adequate number of principal components to represent the process in an "optimal" way [162]. If fewer principal components are selected than required, a poor model will be obtained which has an incomplete representation of the process. On the contrary, if more principal components than necessary are selected, the model will be over-parameterized and will include a significant amount of noise. Based on the available literature, a simple but reliable CPV (Cumulative Percent Variance) method has been chosen for our application.

   The CPV [98] is a measure of the percent variance captured by the first l PCs:

$$CPV(l) = 100 \left(\frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \right) \% \tag{2.66}$$

   where p is the number of variables and λ_i is the i^{th} eigenvalue of the covariance

matrix S or the correlation matrix R of the data matrix X. Usually one selects a desired CPV, e.g. 90%. The criterion of CPV method is somewhat arbitrary and it tends to select a large number of principal components when the process is very noisy.

2. **Determining the Control Limits for the T^2 & Q Statistics**

Two important underlying assumptions of PCA are that the data analyzed are multivariate normally distributed and independent. The calculation of the control limits for the T^2 and Q statistics is then based on the distribution of the T^2 and Q scores with the significance level of these statistics. The actual control limits also depend on the reference data information, e.g., the number of observations, the number of variables, eigenvalues of the sample covariance or correlation matrix, etc.

- **Control limit for T^2 statistic**

  Given a vector of measurements x on p normally distributed variables, with an in-control covariance matrix Σ, one can test whether the current mean of the multivariate process is at its population mean μ by computing the statistic $\chi^2 = (x - \mu)\Sigma^{-1}(x - \mu)^T$. This statistic is used when the sample size n is large and the distribution will follow a centered chi-squared distribution with p degrees of freedom if the mean is equal to μ. The control limit given the significance level α is $\chi^2_{\alpha,n}$.

  When the in-control covariance matrix Σ is unknown and must be es-

timated from a limited amount of data, it is more appropriate to use Hotelling's T^2 statistic given by $T^2 = (x-\mu)S^{-1}(x-\mu)^T$, where S (the sample covariance) is an estimate of Σ. The control limit is obtained using the F-distribution [153] and will depend on the degrees of freedom available for the estimate S.

For a single new multivariate observation ($p \times 1$) and an estimate S based on n past multivariate observations, the exact distribution of the T^2 score can be approximated by $\frac{p(n^2-1)}{n(n-p)}F_\alpha(p, n-p)$ and the exact control limits given the significance level α are:

$$\begin{cases} LCL = \frac{p(n^2-1)}{n(n-p)}F_{1-\alpha}(p, n-p) \\ UCL = \frac{p(n^2-1)}{n(n-p)}F_\alpha(p, n-p) \end{cases} \tag{2.67}$$

LCL – lower control limit; UCL – upper control limit

Establishing Control in the Start-up Stage:

In the start-up stage when the sample size is small and the transients make it difficult for the process to have a stable distribution, neither of the two above assumptions holds true. The Hotelling's T^2 statistic has a beta distribution $T^2 \backsim \frac{(n-1)^2}{n}B(\frac{p}{2}, \frac{n-p-1}{2})$ and the upper control limit is

$$\frac{(n-1)^2}{n} \cdot \frac{p/(n-p-1)F_\alpha(p, n-p-1)}{1 + p/(n-p-1)F_\alpha(p, n-p-1)} \tag{2.68}$$

Note: The control limit of the start-up stage using the exact distribution (beta) is more stringent than using the approximations (F and chi-squared). The control limit is valid only when the individual samples collected in the start-up stage of the process are checked to see whether they

fall within the control limit. It is useful to filter out any abrupt changed and out-of-limit samples due to the transient behavior.

- **Control limit for the Q statistic**

  Let x be an observation from a multinormal population $N(0, S)$ and λ_i, $i = 1, 2, \ldots, p$ denote the eigenvalues of its sample covariance S, then approximations to the control limit [67] for the Q statistic $Q = x^T x$ with significance level α are given by:

$$C_\alpha = \theta_1[1 - \theta_2 h_0(1 - h_0)/\theta_1^2 + z_\alpha(2\theta_2 h_0^2)^{1/2}/\theta_1]^{1/h_0} \tag{2.69}$$

  where,

$$\theta_1 = \sum_{i=1}^{p} \lambda_i, \theta_2 = \sum_{i=1}^{p} \lambda_i^2, \theta_3 = \sum_{i=1}^{p} \lambda_i^3, h_0 = 1 - 2\theta_1\theta_3/3\theta_2^2 \tag{2.70}$$

  For a reference data set X $(n \times p)$, n is the number of observations and p is the number of variables, its covariance matrix is S and λ_i, $i = 1, 2, \ldots, p$ are the eigenvalues of S. Let q be the number of PCs, then the dimension of the residual subspace is $p - q$. λ_i, $i = 1, 2, \ldots, q$ are the eigenvalues corresponding to the PC subspace and λ_j, $j = q+1, \ldots, p$ are the eigenvalues corresponding to the residual subspace.

  For any single observation x, let $\widehat{x}$ denote the projection onto the PC subspace, then $\widetilde{x} = x - \widehat{x}$ denotes the projection onto the residual subspace. When calculating the control limit for the Q statistic, now we are looking at the quadratic form $Q = \widetilde{x}^T\widetilde{x}$, and the calculation of the control limit is

based on the eigenvalues λ_j, $j = q+1, \ldots, p$ which explains the variation of the reference data set in the residual subspace, instead of all p eigenvalues. Now, we use Eq. 2.69 with the parameters in Eq. 2.71 to calculate the control limit of the Q statistic.

$$\theta_1 = \sum_{j=q+1}^{p} \lambda_j, \theta_2 = \sum_{j=q+1}^{p} \lambda_j^2, \theta_3 = \sum_{j=q+1}^{p} \lambda_j^3, h_0 = 1 - 2\theta_1\theta_3/3\theta_2^2 \qquad (2.71)$$

2.4 Simulation Study

2.4.1 Plant Model

A well-known CSTR (Continuously well-Stirred Tank Reactor) model [97] is used for the case studies in the MMEKF work. In this section, the plant model and its implementation are described in detail and the faults that are considered in the study are introduced.

2.4.1.1 Process Description

Fig. 2.16 illustrates the CSTR model in which an irreversible and exothermic reaction $A \Longrightarrow B$ takes place. The reactor is operated by three control loops that regulate the outlet temperature t_o, the inlet flow rate f_i and the tank level l. A cooling jacket surrounds the reactor and the coolant is water in this case. Negligible heat losses, constant densities and perfect mixing inside the tank are assumed. Therefore, the temperature in the jacket is uniform and equal to the outlet temperature. The process variables are as follows.

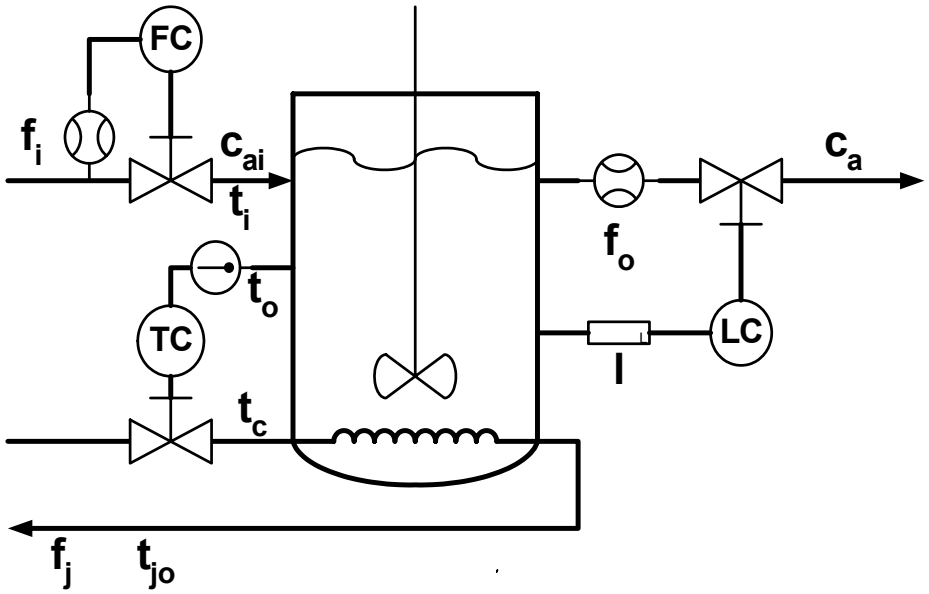

Figure 2.16: Process Flow of CSTR (Luyben Model)

c_{ai}, c_a	concentration of A at the inlet and outlet, respectively;
f_i, f_o	flow rate of the liquid at inlet and outlet, respectively;
f_j	flow rate of coolant;
v	volume of the tank;
t_i	temperature of the inlet reactant;
t_{jo}	temperature of the outlet coolant (water);
t_o	temperature of the tank;
f_{cv}, l_{cv}, t_{cv}	control valve openings.

The parameters, assumed to be constant, are as follows.

k_0	frequency factor;
E_a	activation energy;
R	gas constant;
$\rho \cdot c_p$	volumetric heat capacity;
ΔH	heat of reaction;
a_0	heat exchange area;
u	overall heat transfer coefficient;
a_r	area of the tank;
v_j	jacket volume;
c_j	heat capacity of water;
ρ_j	density of water;
t_c	temperature of the inlet coolant (water).

The equations describing the system dynamics are

Mass balance:

$$\begin{aligned} \frac{dv}{dt} &= f_i - f_o \\ \frac{d(v \cdot c_a)}{dt} &= f_i \cdot c_{ai} - f_o \cdot c_a - v \cdot (k_0 \cdot \exp \frac{-E_a}{R \cdot t_o}) \cdot c_a \end{aligned} \tag{2.72}$$

Energy balance:

$$\begin{aligned} \rho \cdot c_p \cdot \frac{d(v \cdot t_o)}{dt} &= \rho \cdot c_p \cdot (f_i \cdot t_i - f_o \cdot t_o) - \Delta H \cdot v \cdot (k_0 \cdot \exp \frac{-E_a}{R \cdot t_o}) \cdot c_a \\ &\quad -u \cdot a_0 \cdot (t_o - t_{jo}) \\ \rho_j \cdot v_j \cdot c_j \cdot \frac{dt_{jo}}{dt} &= \rho_j \cdot c_j \cdot f_j \cdot (t_c - t_{jo}) + u \cdot a_0 \cdot (t_o - t_{jo}) \end{aligned} \tag{2.73}$$

The system is highly nonlinear. All equations and parameter values are taken from [94].

2.4.1.2 Controller Loops

For computer simulation, the plant model is implemented using Matlab. The basic time unit is an hour. The step size for Euler integration is denoted by dt and is usually 0.001 (h).

Three PI controllers are used to regulate the outlet temperature t_o, the inlet flow rate of the reactant f_i and the tank level l. A block diagram of each control loop is shown in Fig. 2.17. Equal percentage valves are used to control the flow rate of the reactant, coolant and outlet liquid. To simplify the simulation, upstream and downstream pressures are assumed to be constant. A first-order lag is used to model

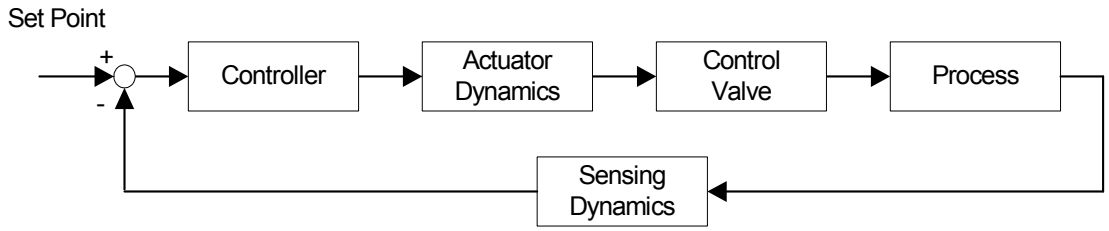

Figure 2.17: Block Diagram of the Control Loop

the actuator and sensing dynamics. The detailed block diagrams of these three PI controllers and their ODEs used for simulation are given below.

1. f_i **PI Control Loop**

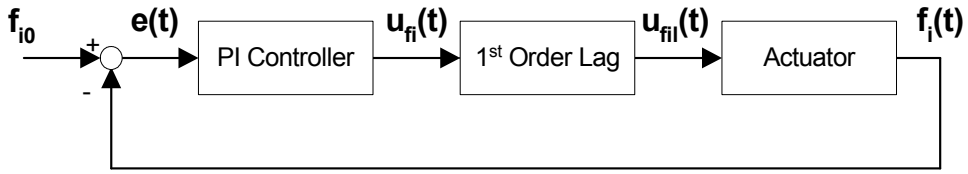

Figure 2.18: Block Diagram of the f_i Control Loop

The block diagram of the f_i PI control loop is shown in Fig. 2.18, where f_{i0} is the set point of the inlet flow rate, $e(t)$ is the error term, $u_{fi}(t)$ is the controller output, $u_{fil}(t)$ is the 1^{st} order lagged controller output. The PI controller output is $u_{fi}(t) = k_p \cdot e(t) + k_i \cdot \int_0^t e(\tau)d\tau$, where $e(t) = f_{i0} - f_i(t)$. The dynamics of the 1^{st} order lag are given by $\dot{u}_{fil}(t) = \frac{G}{\tau}u_{fi}(t) - \frac{1}{\tau}u_{fil}(t)$, where τ is the time lag and G is a constant. The actuator dynamic is $f_i(t) = \frac{Cv}{\alpha}\alpha^{u_{fil}(t)/100}$, where Cv and α are control valve characteristic constants. Therefore, the ODEs for

this control loop are:

$$\begin{cases} \dot{f_i}(t) = \frac{\ln\alpha}{100} \cdot f_i(t) \cdot [\frac{G}{\tau} \cdot u_{fi}(t) - \frac{1}{\tau} \cdot u_{fil}(t)] \\ \dot{u}_{fi}(t) = -k_p \cdot \frac{\ln\alpha}{100} \cdot f_i(t) \cdot [\frac{G}{\tau} \cdot u_{fi}(t) - \frac{1}{\tau} \cdot u_{fil}(t)] - k_i \cdot f_i(t) + k_i \cdot f_{i0} \\ \dot{u}_{fil}(t) = \frac{G}{\tau} \cdot u_{fi}(t) - \frac{1}{\tau} \cdot u_{fil}(t) \end{cases} \tag{2.74}$$

2. f_o **PI Control Loop (Regulating of the Tank Level** l**)**

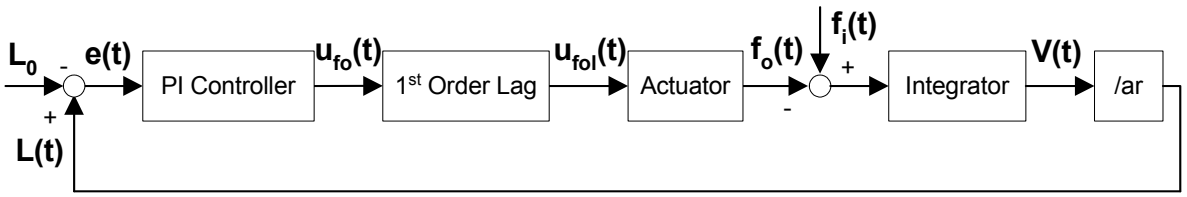

Figure 2.19: Block Diagram of the f_o Control Loop

The block diagram of the f_o PI control loop is shown in Fig. 2.19, where L_0 is the set point of the tank level, $e(t)$ is the error term, $u_{fo}(t)$ is the controller output, $u_{fol}(t)$ is the 1^{st} order lagged controller output, $V(t)$ is the tank volume, and ar is the cross-sectional area of the tank, assuming equal cross-sectional area at any level of the tank. Again, the PI controller output is $u_{fo}(t) = k_p \cdot e(t) + k_i \cdot \int_0^t e(\tau)d\tau$, where $e(t) = L(t) - L_0$. The 1^{st} order lag dynamics are given by $\dot{u}_{fol}(t) = \frac{G}{\tau}u_{fo}(t) - \frac{1}{\tau}u_{fol}(t)$, and the actuator dynamics are given by $f_o(t) = \frac{Cv}{\alpha}\alpha^{u_{fol}(t)/100}$. The tank volume dynamics are given by $\dot{V}(t) = f_i(t) - f_o(t)$, and $L(t) = \frac{V(t)}{ar}$. Similarly, this control loop can be described by

the following ODEs:

$$\begin{cases} \dot{f_o}(t) = \frac{\ln\alpha}{100} \cdot f_o(t) \cdot [\frac{G}{\tau} \cdot u_{fo}(t) - \frac{1}{\tau} \cdot u_{fol}(t)] \\ \dot{u}_{fo}(t) = \frac{k_p}{ar} \cdot [f_i(t) - f_o(t)] + k_i \cdot [\frac{V(t)}{ar} - L_0] \\ \dot{u_{fol}}(t) = \frac{G}{\tau} \cdot u_{fo}(t) - \frac{1}{\tau} \cdot u_{fol}(t) \end{cases} \tag{2.75}$$

3. f_j **PI Control Loop (Regulating the Outlet Temperature** t_o**)**

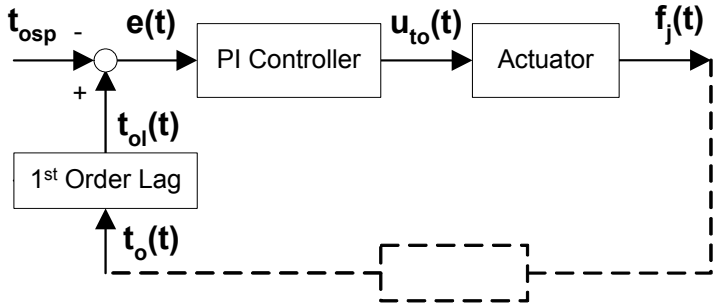

Figure 2.20: Block Diagram of the f_j Control Loop

The block diagram of the f_j PI control loop is shown in Fig. 2.20, where t_{osp} is the set point of the outlet temperature, $e(t)$ is the error term, $u_{to}(t)$ is the controller output, $t_{ol}(t)$ is the 1^{st} order lagged outlet temperature t_o. This control loop can be described by the following ODEs:

$$\begin{cases} \dot{t}_{ol}(t) = \frac{G}{\tau} \cdot t_o(t) - \frac{1}{\tau} \cdot t_{ol}(t) \\ \dot{u}_{to}(t) = k_p \cdot [\frac{G}{\tau} \cdot t_o(t) - \frac{1}{\tau} \cdot t_{o,l}(t)] + k_i \cdot [t_{ol}(t) - t_{osp}] \end{cases} \tag{2.76}$$

The process state space model can be obtained by combining the three PI controller loops Eqs. 2.74, 2.75, 2.76 with the mass balance Eq. 2.72 and the energy balance Eq. 2.73.

2.4.1.3 Fault Modes

Three different types of faults that affect the sensor, actuator, and process are listed in Table 2.1.

Table 2.1: List of Faults Studied

Fault #	Fault Name	Fault Class
#1p	Positive bias of inlet flow rate f_i	malfunction of sensor
#1n	Negative bias of inlet flow rate f_i	
#4p	f_{cv} failure (valve hold: $f_i \uparrow$)	malfunction of actuator
#4n	f_{cv} failure (valve hold: $f_i \downarrow$)	
#9p	High inlet temperature of reactant $t_i \uparrow$	malfunction of process
#9n	Low inlet temperature of reactant $t_i \downarrow$	

The detailed descriptions of these three fault modes are listed below.

1. **Sensor Fault f1p/1n**

   The modelling of the sensor fault f1p/1n and how it can affect the f_i control loop is depicted in Fig. 2.21. It looks almost the same as the f_i control loop except that the sensed inlet flow rate $f_{is}(t)$, which contains the fault, is fed back to the controller.

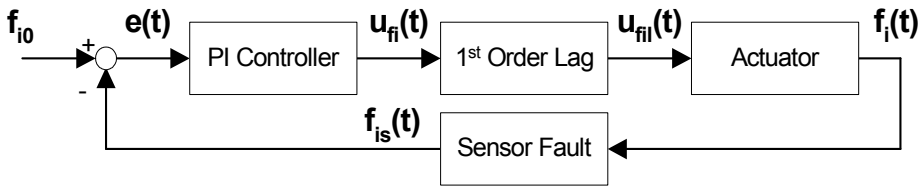

Figure 2.21: Sensor Fault f1p/1n in f_i Control Loop

The sensor fault is described as a step fault (see Eq. 2.25) with certain magni-

tude over the true f_i value with a fault developing time t_r in Eq. 2.77, where f_m is the fault magnitude and the $+/-$ corresponds to f1p/1n, respectively.

$$f_{is}(t) = \begin{cases} f_i(t) & t < t_f \\ f_i(t) \ \pm \frac{f_m}{t_r}(t - t_f) & t_f \leq t \leq t_f + t_r \\ f_i(t) \ \pm f_m & t > t_f + t_r \end{cases} \tag{2.77}$$

The f_i PI control loop model with fault f1p/1n is shown below.

$$\begin{cases} \dot{f_i}(t) = \frac{\ln \alpha}{100} f_i(t)[\frac{G}{\tau} u_{fi}(t) - \frac{1}{\tau} u_{fil}(t)] \\ \dot{u}(t) = \begin{cases} -k_p \dot{f_i}(t) + k_i f_{i0} - k_i f_i(t) & t < t_f \\ -k_p[\dot{f_i}(t) \ \pm \frac{f_m}{t_r}] + k_i f_{i0} - k_i[f_i(t) \ \pm \frac{f_m}{t_r}(t - t_f)] & t_f \leq t \leq t_f + t_r \\ -k_p \dot{f_i}(t) \ + k_i f_{i0} - k_i[f_i(t) \ \pm f_m] & t > t_f + t_r \end{cases} \\ \dot{u_{fil}}(t) = \frac{G}{\tau} u_{fi}(t) - \frac{1}{\tau} u_{fil}(t) \end{cases} \tag{2.78}$$

2. **Actuator Fault f4p/4n**

The modelling of the actuator fault f4p/4n and how it can affect the f_i control loop is depicted in Fig. 2.22, where $f_{ia}(t)$ denotes the normal inlet flow rate after the actuator and $f_i(t)$ is contaminated by actuator fault.

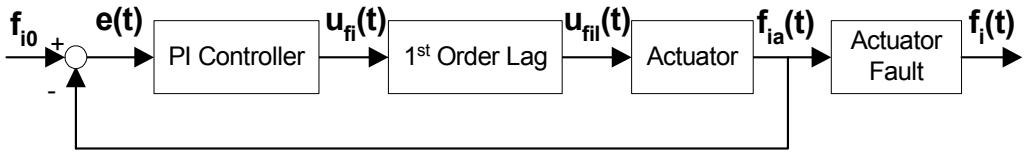

Figure 2.22: Actuator Fault f4p/4n in f_i Control Loop

The actuator fault is also modelled as a step fault, with an additional actuator

time lag t_{lag}.

$$f_i(t) = \begin{cases} f_{ia}(t) & t < t_f + t_{lag} \\ f_{ia}(t) \pm \frac{f_m}{t_r}(t - t_f - t_{lag}) & t_f + t_{lag} \le t \le t_f + t_r + t_{lag} \\ f_{ia}(t) \pm f_m & t > t_f + t_r + t_{lag} \end{cases} \tag{2.79}$$

The f_i PI control loop model with fault f4p/4n is shown below.

$$\begin{cases} \dot{f}_{ia}(t) = \frac{\ln \alpha}{100} f_{ia}(t)[\frac{G}{\tau} u_{fi}(t) - \frac{1}{\tau} u_{fil}(t)] \\ \dot{u}(t) = -k_p \dot{f}_{ia}(t) + k_i f_{i0} - k_i f_{ia}(t) \\ \dot{f}_i(t) = \begin{cases} \dot{f}_{ia}(t) & t < t_f + t_{lag} \\ \dot{f}_{ia}(t) \pm \frac{f_m}{t_r} & t_f + t_{lag} \le t \le t_f + t_r + t_{lag} \\ \dot{f}_{ia}(t) & t > t_f + t_r + t_{lag} \end{cases} \\ \dot{u_{fil}}(t) = \frac{G}{\tau} u_{fi}(t) - \frac{1}{\tau} u_{fil}(t) \end{cases} \tag{2.80}$$

3. **Process Fault f9p/9n**

   The process fault is modelled as a step change in the process parameter. For example, fault f9p/9n is modelled as a step increase/decrease of t_i, the temperature of the inlet reactant. Unlike the sensor and actuator faults, the process fault is propagated throughout the entire simulation process.

2.4.2 EKF FDI System Simulation

The detailed block diagram of the EKF fault diagnosis scheme is shown in Fig. 2.23, where $x(t)$ is the system state, $u(t)$ is the input, y_k is the true measurement and z_k is the real measurement at time t_k. To make the EKF model and the process

model match when they are under the same operating condition, the EKF model and the process model should be completely parallel, i.e. they should have exactly the same structure with regard to the actuator, sensing and process dynamics. They also share the same controller information which in the real situation, is obtained from the measurements. Under sensor or actuator fault condition, the sensing dynamics or actuator dynamics of either the process model or the EKF model could be modified to accommodate the faulty condition.

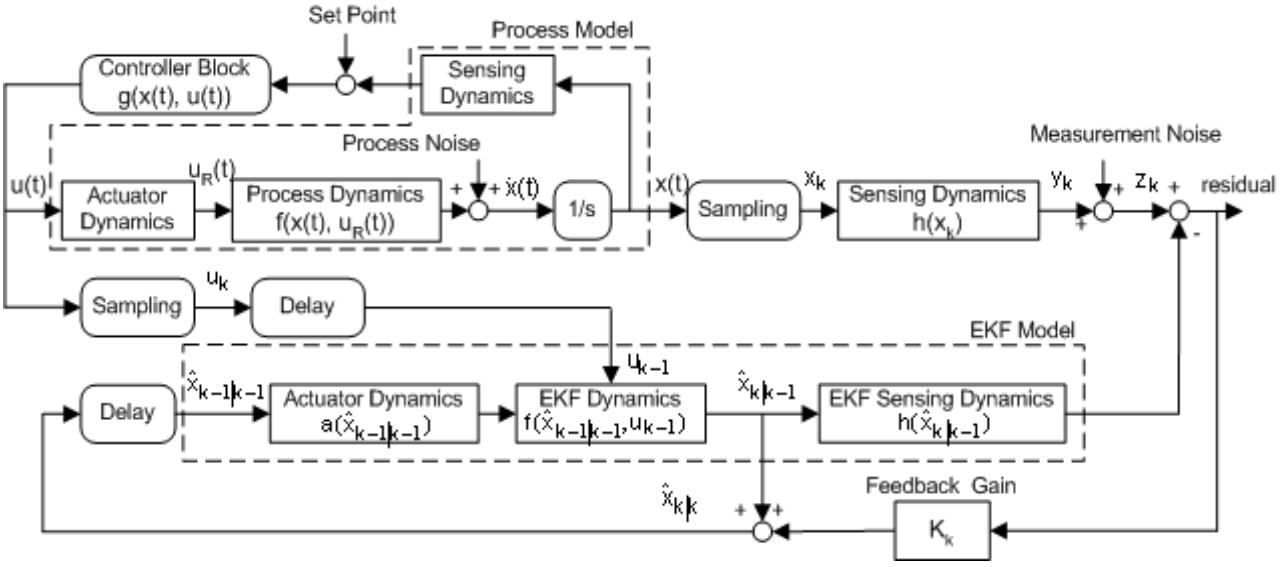

Figure 2.23: Detailed Block Diagram of the EKF FDI System

2.4.2.1 Application of EKF to the CSTR Process

The EKF is used to estimate the unmeasurable state variables and relevant parameters in the CSTR process. The system and measurement equations are given by Eq. 2.20, the situation where the process is continuous and the measurement is discrete.

System Dynamics Define the state vector

$$\begin{aligned} x &= [v, v \cdot c_a, v \cdot t_o, t_{jo}, f_i, f_o]^T \\ &= [x_1, x_2, x_3, x_4, x_5, x_6]^T \end{aligned} \tag{2.81}$$

For computational reasons, the system equations are modified to be dimensionless. The normalized state variables are defined as shown in Eq. 2.82, where $v_s = 50(ft^3)$, $c_{as} = 0.2447$, $t_{os} = 600(R)$, $t_{jos} = 594.6(R)$, $f_{is} = 40(ft^3/h)$, $f_{os} = 40(ft^3/h)$, and these are the steady-state values of the process.

$$\begin{aligned} x_N &= [x_{N1}, x_{N2}, x_{N3}, x_{N4}, x_{N5}, x_{N6}]^T \\ &= [\frac{x_1 - v_s}{v_s}, \frac{x_2 - v_s \cdot c_{as}}{v_s \cdot c_{as}}, \frac{x_3 - v_s \cdot t_{os}}{v_s \cdot t_{os}}, \frac{x_4 - t_{jos}}{t_{jos}}, \frac{x_5 - f_{is}}{f_{is}}, \frac{x_6 - f_{os}}{f_{os}}]^T \end{aligned} \tag{2.82}$$

To simplify the notation, the 'N' is omitted from here on, and because the state variables are defined as deviation variables from their steady-state values, the initial state is given as $N(0, P_0)$. $\omega(t) = [\omega_v, \omega_{vca}, \omega_{vto}, \omega_{tjo}, \omega_{fi}, \omega_{fo}]^T$ represents the modelling uncertainty. From Eqs. 2.72, 2.73, and 2.74, the system equations of the CSTR are written in vector form as shown in Eq. 2.83.

$$\frac{dx}{dt} = \begin{bmatrix} c_1(x_5+1) - c_2(x_6+1) \\ c_3(x_5+1) - c_4\frac{(x_6+1)(x_2+1)}{(x_1+1)} - c_5(x_2+1)\exp(\frac{-c_6(x_1+1)}{(x_3+1)}) \\ c_7(x_5+1) - c_8\frac{(x_6+1)(x_3+1)}{(x_1+1)} - c_9(x_2+1)\exp(\frac{-c_6(x_1+1)}{(x_3+1)}) \\ -c_{10}\frac{(x_3+1)}{(x_1+1)} + c_{11}(x_4+1) \\ -c_{12}f_j(t)(x_4+1) - c_{13}(x_4+1) + c_{14}\frac{(x_3+1)}{(x_1+1)} + c_{15}f_j(t) \\ c_{16}\cdot u_{fi}(t)(x_5+1) - c_{17}\cdot u_{fil}(t)(x_5+1) \\ c_{18}\cdot u_{fo}(t)(x_6+1) - c_{19}\cdot u_{fol}(t)(x_6+1) \end{bmatrix} + \begin{bmatrix} \frac{\omega_v}{v_s} \\ \frac{\omega_{vca}}{v_s\cdot c_{as}} \\ \frac{\omega_{vto}}{v_s\cdot t_{os}} \\ \\ \frac{\omega_{tjo}}{t_{jos}} \\ \frac{\omega_{fi}}{f_{is}} \\ \frac{\omega_{fo}}{f_{os}} \end{bmatrix} \tag{2.83}$$

where

$$\begin{aligned} c_1 &= \frac{f_{is}}{v_s}, c_2 = \frac{f_{os}}{v_s}, c_3 = \frac{c_{ai}\cdot f_{is}}{v_s\cdot c_{as}}, c_4 = \frac{f_{os}}{v_s}, c_5 = k_0, c_6 = \frac{E_a}{R\cdot t_{os}}, c_7 = \frac{t_i\cdot f_{is}}{v_s\cdot t_{os}} \\ c_8 &= \frac{f_{os}}{v_s}, c_9 = \frac{\Delta H\cdot k_0\cdot c_{as}}{t_{os}\cdot\rho\cdot c_p}, c_{10} = \frac{\mu\cdot a_0}{v_s\cdot\rho\cdot c_p}, c_{11} = \frac{\mu\cdot a_0\cdot t_{jos}}{v_s\cdot t_{os}\cdot\rho\cdot c_p} \\ c_{12} &= \frac{1}{v_j}, c_{13} = \frac{\mu\cdot a_0}{\rho\cdot v_j\cdot c_j}, c_{14} = \frac{\mu\cdot a_0\cdot t_{os}}{\rho\cdot v_j\cdot c_j\cdot t_{jos}}, c_{15} = \frac{t_c}{v_j\cdot t_{jos}} \\ c_{16} &= \frac{\ln\alpha\cdot G_1}{100\cdot\tau_1}, c_{17} = \frac{\ln\alpha}{100\cdot\tau_1}, c_{18} = \frac{\ln\alpha\cdot G_2}{100\cdot\tau_2}, c_{19} = \frac{\ln\alpha}{100\cdot\tau_2} \end{aligned} \tag{2.84}$$

and τ_1/G_1 and τ_2/G_2 are the time lags/constants of the f_i and f_o PI control loops, respectively.

Define the input vector

$$\begin{aligned} u &= [u_{fi}, u_{fil}, u_{fo}, u_{fol}, u_{to}, t_{ol}, f_j]^T \\ &= [u_1, u_2, u_3, u_4, u_5, u_6, u_7]^T \end{aligned} \tag{2.85}$$

The controller dynamics $g(x(t), u(t))$ are shown in Eq. 2.86.

$$\frac{du}{dt} = \begin{bmatrix} -c_{20}(x_5+1)u_1 + c_{21}(x_5+1)u_2 - c_{22}(x_5+1) + c_{23} \\ c_{24}\cdot u_1 - c_{25}\cdot u_2 \\ c_{26}(x_5+1) - c_{27}(x_6+1) + c_{28}(x_1+1) - c_{29} \\ c_{30}\cdot u_3 - c_{31}\cdot u_4 \\ c_{32}\frac{(x_3+1)}{(x_1+1)} + c_{33}\cdot u_6 - c_{34} \\ c_{35}\frac{(x_3+1)}{(x_1+1)} - c_{36}\cdot u_6 \\ c_{37}\frac{(x_3+1)}{(x_1+1)}u_7 + c_{38}\cdot u_6\cdot u_7 - c_{39}\cdot u_7 \end{bmatrix} \tag{2.86}$$

where

$$\begin{aligned} c_{20} &= \frac{f_{is}\cdot K_{p1}\cdot \ln\alpha\cdot G_1}{100\cdot\tau_1}, c_{21} = \frac{f_{is}\cdot K_{p1}\cdot\ln\alpha}{100\cdot\tau_1}, c_{22} = f_{is}\cdot K_{i1}, c_{23} = K_{i1}\cdot f_{i0} \\ c_{24} &= \frac{G_1}{\tau_1}, c_{25} = \frac{1}{\tau_1}, c_{26} = \frac{f_{is}\cdot K_{p2}}{ar}, c_{27} = \frac{f_{os}\cdot K_{p2}}{ar}, c_{28} = \frac{v_s\cdot K_{i2}}{ar} \\ c_{29} &= K_{i2}\cdot L_0, c_{30} = \frac{G_2}{\tau_2}, c_{31} = \frac{1}{\tau_2}, c_{32} = \frac{t_{os}\cdot K_{p3}\cdot G_3}{\tau_3}, c_{33} = K_{i3} - \frac{K_{p3}}{\tau_3} \\ c_{34} &= K_{i3}\cdot t_{0sp}, c_{35} = \frac{t_{os}\cdot G_3}{\tau_3}, c_{36} = \frac{1}{\tau_3}, c_{37} = \frac{t_{os}\cdot K_{p3}\cdot G_3\cdot\ln\alpha}{100\cdot\tau_3} \\ c_{38} &= (K_{i3} - \frac{K_{p3}}{\tau_3})\frac{\ln\alpha}{100}, c_{34} = \frac{K_{i3}\cdot t_{0sp}\cdot\ln\alpha}{100} \end{aligned} \tag{2.87}$$

Detailed Simulation Procedure Computation is performed by repeating the following propagation and updating steps in a discrete manner.

1. **Prediction (Approximation):**

$$\widehat{x}_{k|k-1} = \widehat{x}_{k-1|k-1} + f(\widehat{x}_{k-1|k-1}, u_{k-1}, t_{k-1})\cdot dt \tag{2.88}$$

The continuous process state space equation is first discretized and an Euler approximation is used in Eq. 2.88, where dt is the sampling interval, $\widehat{x}_{k|k-1}$ is

the estimated state at t_k, assuming that the state estimate $\widehat{x}_{k-1|k-1}$ and input u_{k-1} at time t_{k-1} are known. Higher order approximations can also be used if a more precise prediction model is needed.

2. **Propagation of the Error Covariance:**

The error covariance dynamics are given in Eq. 2.89 and the propagation of error covariance is given in Eq. 2.90.

$$\frac{d}{dt}P(t) = F(\widehat{x}(t), u(t), t) \cdot P(t) + P(t) \cdot F^T(\widehat{x}(t), u(t), t) + Q(t) \tag{2.89}$$

$$\begin{aligned} P_{k|k-1} &= P_{k-1|k-1} + [F(\widehat{x}_{k-1|k-1}, u_{k-1}, t_{k-1}) \cdot P_{k-1|k-1} + \\ &\quad P_{k-1|k-1} \cdot F^T(\widehat{x}_{k-1|k-1}, u_{k-1}, t_{k-1}) + Q(t_{k-1})] \cdot dt \end{aligned} \tag{2.90}$$

where

$F(\widehat{x}(t), u(t), t) = \frac{\partial f}{\partial x} \mid_{x=\widehat{x}}$	the Jacobian matrix of f evaluated at $\widehat{x}$;
$P_{k\|k-1}$	the estimated error covariance at time t_k
$P_{k-1\|k-1}$	the error covariance matrix of at time t_{k-1}.

The function f is linearized at $x = \widehat{x}(F(\widehat{x}(t), u(t), t))$ at every time step.

3. **Updating the Actual State:**

$$x_k = x_{k-1} + [f(x_{k-1}, u_{k-1}, t_{k-1}) + Q^{\frac{1}{2}} \cdot \omega(t_{k-1})] \cdot dt \tag{2.91}$$

4. **Calculation of the Actual Output:**

$$z_k = h_k(x_k) + R^{\frac{1}{2}} \cdot \upsilon_k \tag{2.92}$$

5. **Updating the State Estimate and the Error Covariance:**

The measurement z_k is taken at $t = t_k$ and used to update $\widehat{x}_{k|k-1}$ and $P_{k|k-1}$ to give $\widehat{x}_{k|k}$ and $P_{k|k}$ according to the update formulas:

$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k \cdot [z_k - H_k(\widehat{x}_{k|k-1}) \cdot \widehat{x}_{k|k-1}] \tag{2.93}$$

$$P_{k|k} = [I - K_k \cdot H_k(\widehat{x}_{k|k-1})] \cdot P_{k|k-1} \tag{2.94}$$

where

$$H_k(\widehat{x}_{k|k-1}) = \frac{\partial h_k(x)}{\partial x}\Big|_{x=\widehat{x}_{k|k-1}} \tag{2.95}$$

$$K_k = P_{k|k-1} \cdot H_k^T(\widehat{x}_{k|k-1}) \cdot \left[H_k(\widehat{x}_{k|k-1}) \cdot P_{k|k-1} \cdot H_k^T(\widehat{x}_{k|k-1}) + R\right]^{-1} \tag{2.96}$$

6. $k = k + 1$, **go back to step 1**

Observability Issue The model (Eq. 2.83) has unmeasured state variables and parameters that can change during different operating modes of the process. If parameters such as μ, k_0, and c_{ai} can be estimated, it is very helpful for fault detection and diagnosis. However, because they are not measurable directly and are necessary for the detection of certain faults, the state vector is augmented to include these variables and parameters to be estimated by the EKF. The augmented state must be observable, otherwise the change (fault) in the process will not be shown in the measurement correctly and it is possible under this circumstance that the measurement under a fault condition will not be any different than the normal condition. Various realizations were considered and only the observable one is selected in the simulation study.

After trial and error, $[v, v \cdot c_a, v \cdot t_o, t_{jo}, f_i, f_o]^T$ are selected as the state vector. Consider the linearized state space model, (F, H) is the observability pair. The Jacobian of Eq. 2.83 is of the following form

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 0 & f_{15} & f_{16} \\ f_{21} & f_{22} & f_{23} & 0 & f_{25} & f_{26} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} & f_{36} \\ f_{41} & 0 & f_{43} & f_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & f_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & f_{66} \end{bmatrix}$$

It is easy to see that when $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$, the observability matrix is of full rank. Therefore, v, $v \cdot t_o$, t_{jo}, f_i, and f_o are the measurable variables and the output equation is written as

$$z_k = H \cdot x(t_k) + \upsilon_k = \begin{bmatrix} x_1 + \upsilon_1 \\ x_3 + \upsilon_2 \\ x_4 + \upsilon_3 \\ x_5 + \upsilon_4 \\ x_6 + \upsilon_5 \end{bmatrix}$$

where $\upsilon_k = [\ \upsilon_1 \quad \upsilon_2 \quad \upsilon_3 \quad \upsilon_4 \quad \upsilon_5\ ]^T$ is the measurement noise vector.

EKF Tuning and Simulation for Fault Detection The parameters for tuning the EKF are Q, R, and P_0, which are assumed to be diagonal matrices. Also, the initial state $\widehat{x}_0$ must be specified. As described in Eq. 2.20, Q, R, and P_0 can be determined from the known noise covariances. After that, however, further tuning by trial and error is always necessary to achieve stable and accurate estimation. Each element of R is inversely proportional to the gain matrix K. Hence, smaller elements are chosen as long as the estimation process is stable to achieve fast convergence. As each element of Q increases, faster response is obtained but the amplitude of fluctuation increases. Conversely, the smaller each element of Q is, the slower the response and the smaller the fluctuation. The elements of Q that correspond to unmeasurable states need to be chosen so that the estimated states track the true values. As long as $\widehat{x}_0$ is given as $N(0, P_0)$, and Q and R are chosen appropriately, it is not necessary to adjust P_0.

2.4.3 Simulation Results

2.4.3.1 Simulation Settings

The admissible fault magnitude range of faults in the simulation study is listed in Table 2.2.

To set up the reference PCA model, the fault magnitude is selected around its mean value as shown in Table 2.3.

Table 2.2: The Admissible Fault Magnitude Range

Fault #	**Fault Name**	**Fault Ratio Range**
#1p	Positive bias of inlet flow rate f_i	[0.02, 0.8]
#1n	Negative bias of inlet flow rate f_i	[0.04, 3.5]
#4p	f_{cv} failure (valve hold: $f_i \uparrow$)	[0.001, 1.3]
#4n	f_{cv} failure (valve hold: $f_i \downarrow$)	[0.001, 0.7]
#9p	High inlet temperature of reactant $t_i \uparrow$	[0.01, 2.2]
#9n	Low inlet temperature of reactant $t_i \downarrow$	[0.12, 0.25]

Table 2.3: The Fault Magnitude Used to Set Up Reference PCA Model

Fault #	**#1p**	**#1n**	**#4p**	**#4n**	**#9p**	**#9n**
Fault Ratio	0.4	1.8	0.65	0.35	1.1	0.185

This choice is justified by the observation that residuals under different fault magnitudes, when the EKF model matches the real process, are all zero-mean and their correlation structures are similar, which is illustrated in Table 2.4. The test is based on the reference PCA model for fault f1p, where both process and EKF fault magnitude are $0.4 * f_{is}$. There are 5 variables in the residual signal and the first 2 PCs are selected, and the correlation PCA is used in the test. Two measures are used to test the similarity between the different correlation structures. The first one measures the change of the subspace [71] spanned by the first 2 PCs between the reference PCA model and the test PCA model, and is denoted by A_{1-2}. Its value ranges from 0 to 1. The smaller A_{1-2}, the more similar the test model is to the reference model. The second one measures the common principal components (CPC) of the two covariance (correlation) matrices, using Krzanowski's idea [82]. The idea is to find the minimum angle, δ, between the subspace defined by the first q columns of P_{1q} and P_{2q}, where

P_{1q} and P_{2q} represent the PCs of the correlation matrices of the reference model and the test model, respectively, and q is 2 in this test. The EKF fault magnitude is set to match the process fault magnitude.

Table 2.4: The Residuals Correlation Structure Similarity Test

Test Configuration	$\mathbf{A_{1-2}}$	$\boldsymbol{\delta}(^\circ)$
process #n & EKF #n	0.018054	0.07655
process #1p & EKF #1p, fault ratio=0.02	0.016045	0.04711
process #1p & EKF #1p, fault ratio=0.8	0.18556	0.02245
process #1p & EKF #1p, fault ratio=0.6	0.14838	0.11487
process #1n & EKF #1n, fault ratio=0.4	0.076784	0.06511
process #4p & EKF #4p, fault ratio=0.65	0.17291	0.20657
process #4n & EKF #4n, fault ratio=0.65	0.17876	0.607
process #9p & EKF #9p, fault ratio=1.1	0.036967	0.22049
process #9n & EKF #9n, fault ratio=0.2	0.17119	0.34074

2.4.3.2 MMEKF FDI Test Results

The MMEKF simulation was conducted for the process under the normal condition and all 6 faulty conditions, fault #1p/1n, 4p/4n, and 9p/9n. The reference fault magnitudes in Table 2.3 were used for both the process and EKF simulations. Under each process condition, 7 (1 normal plus 6 faulty conditions) runs for different EKF types were carried out. Since the test results of positive/negative fault pairs 1p/1n, 4p/4n and 9p/9n show similar pattern, only the results of one fault are shown for each pair. To achieve better visual effect, the y-axis of most graphs is shown with a logarithmic scale. Both the T^2 and Q statistics are plotted against the 99% control limits. The step size is 0.002(h), which provides the best EKF performance in combination with the process simulation. The process simulation starts under normal

condition and the fault is introduced at time $t = 4(h)$ while the EKF simulation of any type starts at $t = 0$. The total simulation time is $8(h)$. The transients of the CSTR process are truncated in most of graphs to better view the interesting features of the MMEKF. That is why the time frame for most graphs is [3, 8]. Except for the case when the process is under normal condition, the transient behavior after a fault is introduced at $t = 4(h)$ is evident.

1. **Process Under Normal Condition**

   The T^2 and Q statistics when the process is under normal condition and the EKF is under normal condition and the 6 fault conditions are given in Fig. 2.24 and 2.25, where the T^2 and Q scores are shown as dashed lines and the 99% control limits are given as solid lines. It is easy to see that only under the matching EKF type (#n), the T^2 and Q scores stay below the control limits. Fault detection and isolation can be achieved in just one step.

2. **Process Under Sensor Fault (f1p)**

   The T^2 and Q statistics when the process is under sensor fault #1p and the EKF is under normal condition and 6 fault conditions are given in Fig. 2.26 and 2.27. We see that when the EKF type is #1p, T^2 and Q scores do not go below control limits until $t \simeq 4.4(h)$. This is understandable since the process is under normal condition before $t = 4(h)$ and the step fault f1p introduces additional transients after $t = 4(h)$. The EKF estimation stabilizes and converges to the actual values until $t \simeq 4.4(h)$.

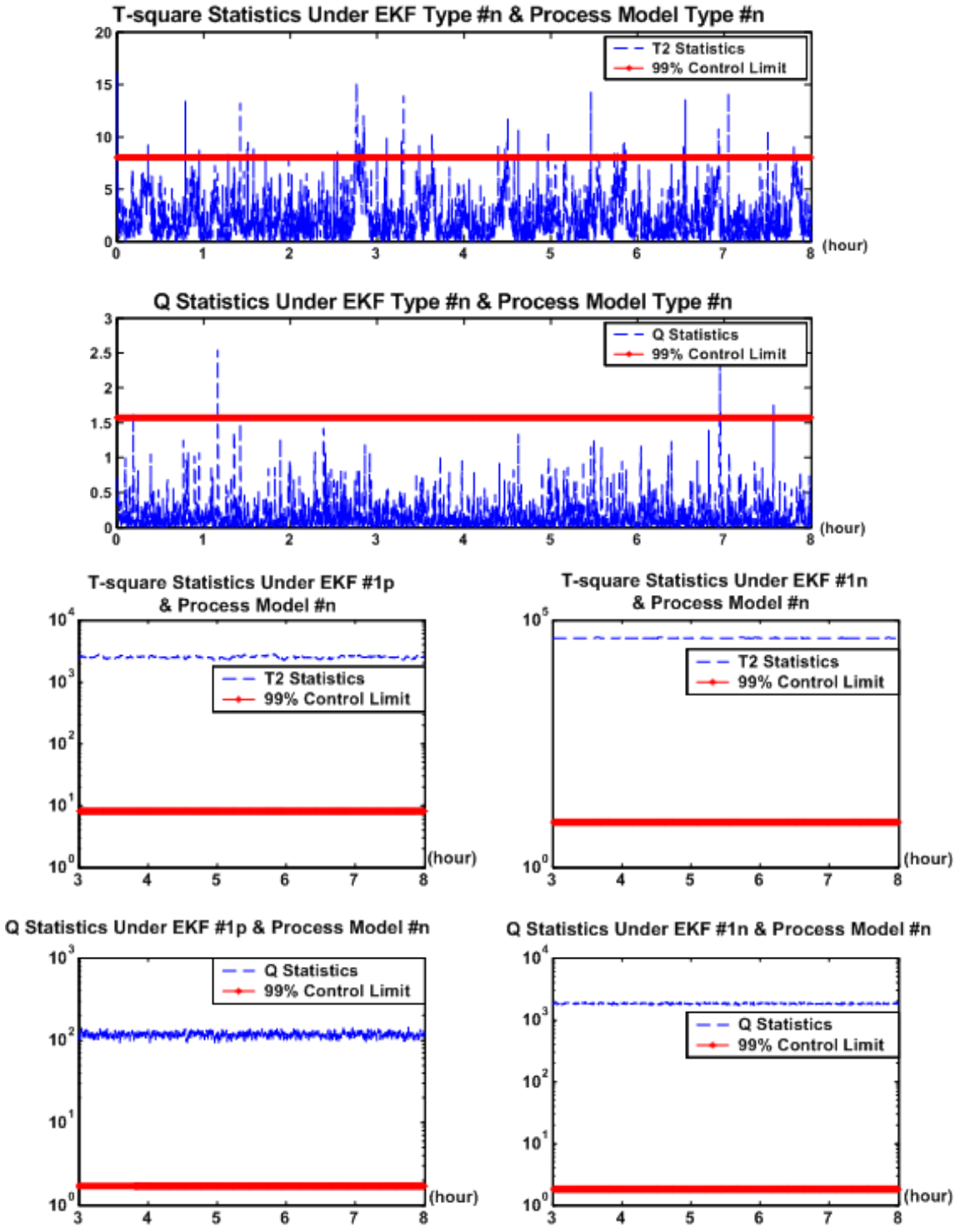

Figure 2.24: T2/Q Statistics of Normal Process (EKF Type n, 1p/1n)

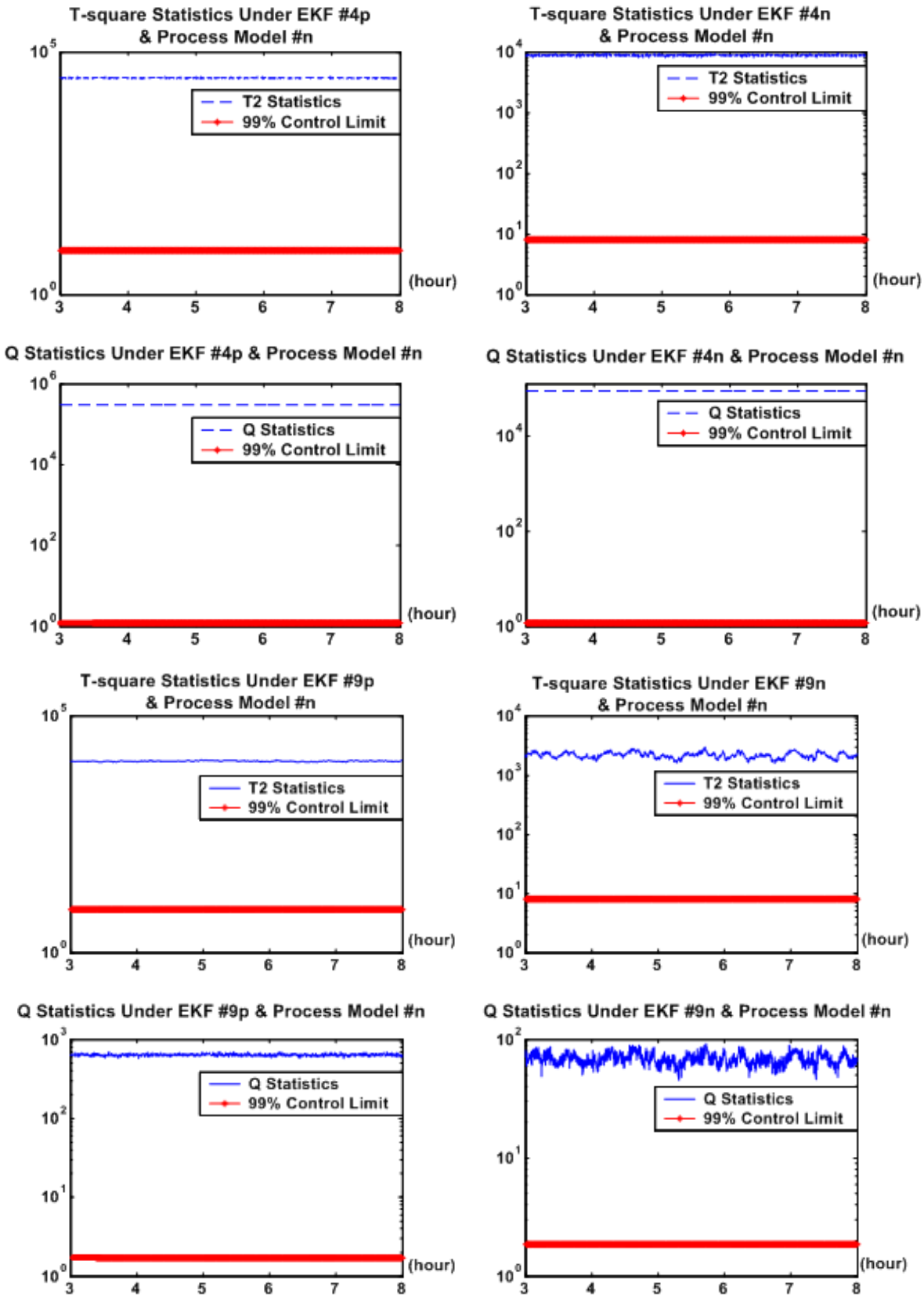

Figure 2.25: T2/Q Statistics of Normal Process (EKF Type 4p/4n, 9p/9n)

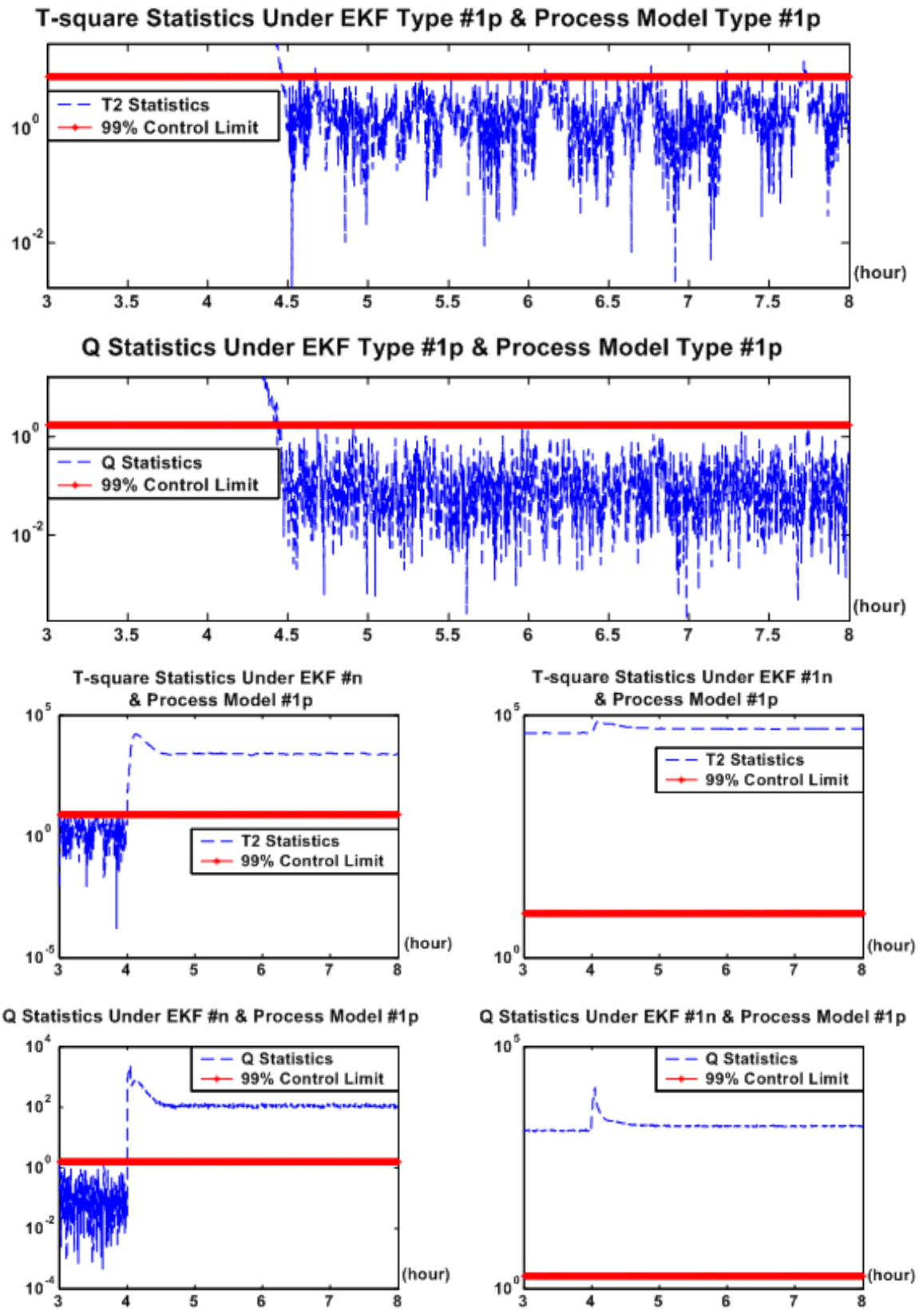

Figure 2.26: T2/Q Statistics of Sensor Fault 1p (EKF Type 1p, n/1n)

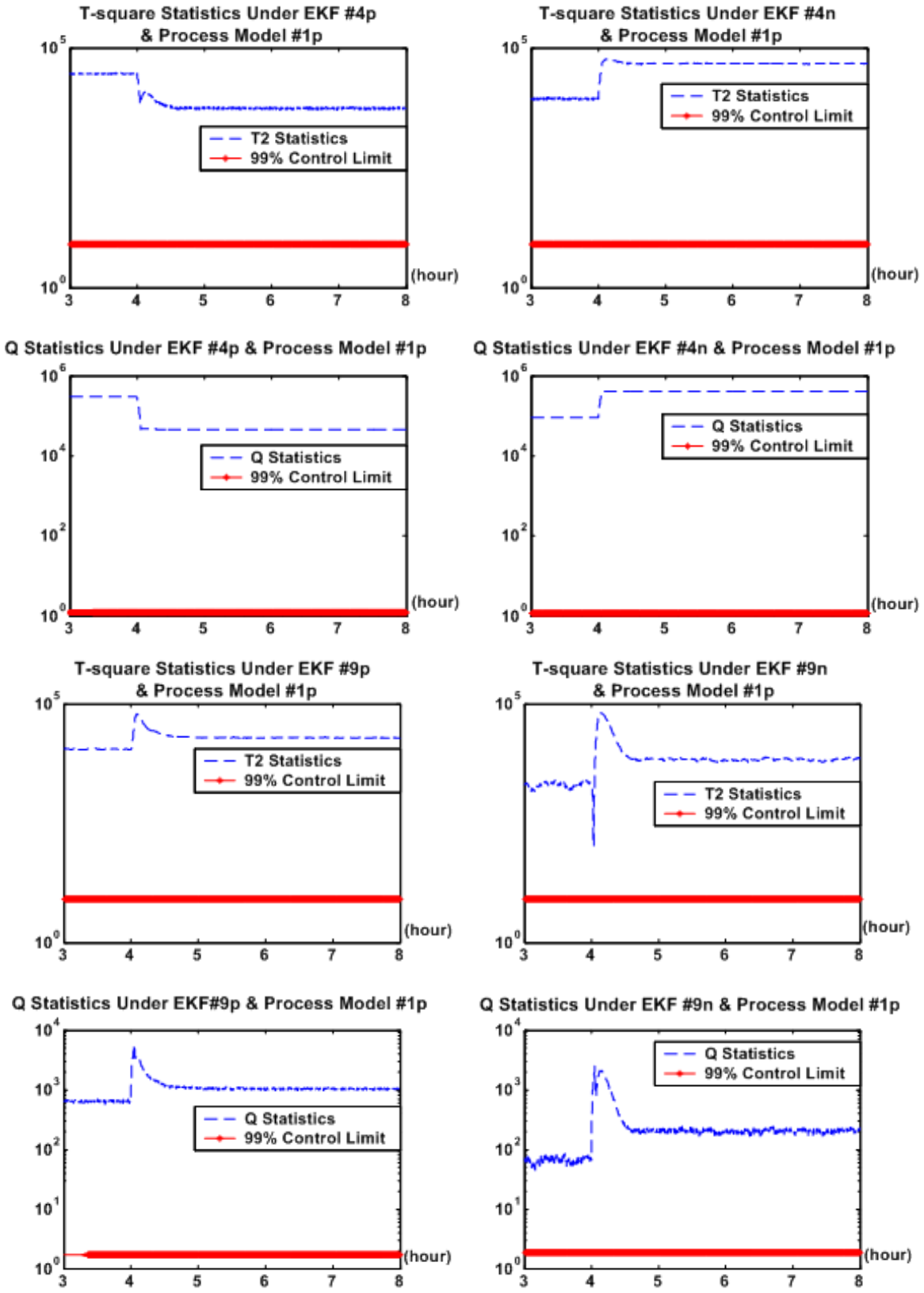

Figure 2.27: T2/Q Statistics of Sensor Fault 1p (EKF Type 4p/4n, 9p/9n)

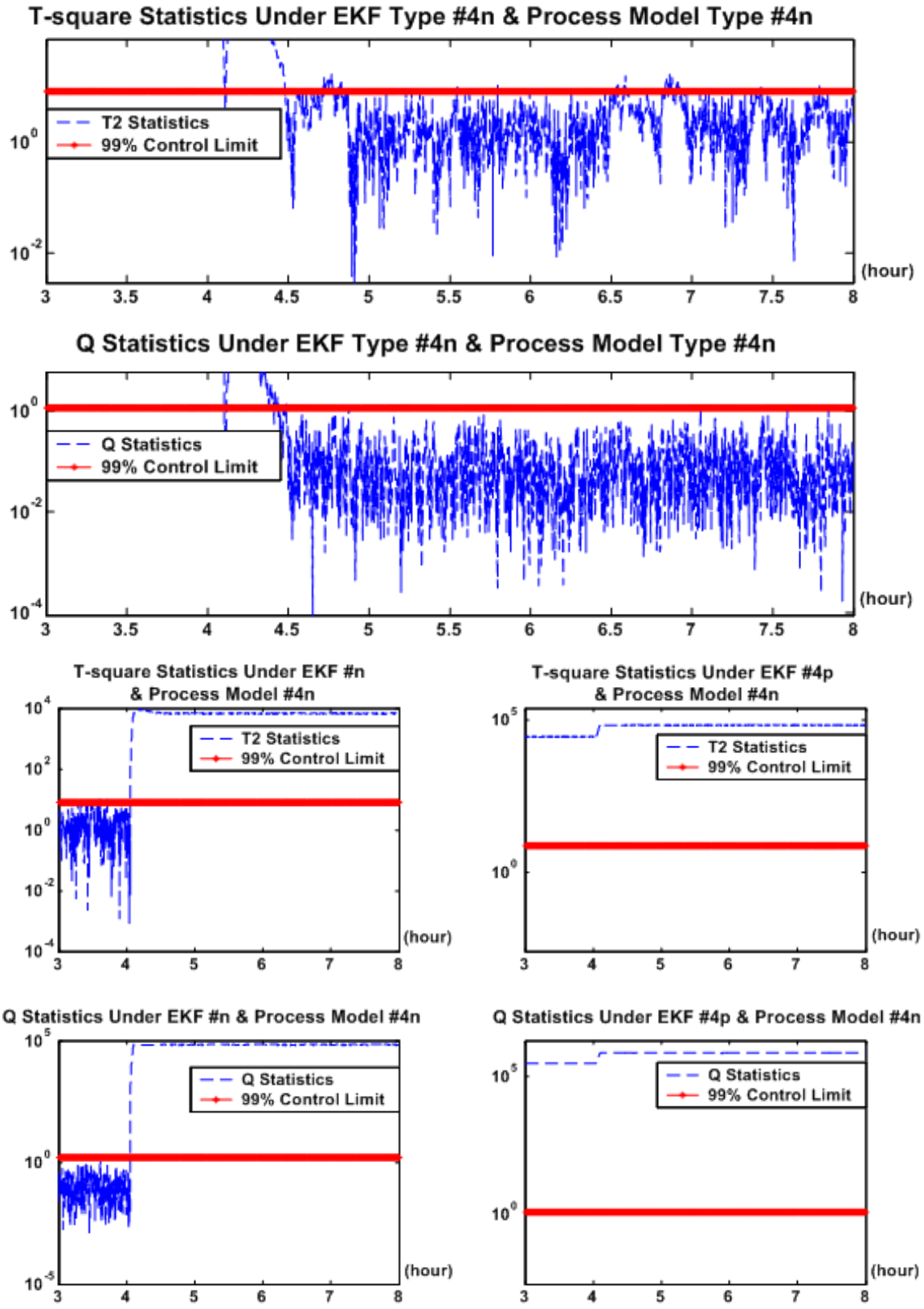

Figure 2.28: T2/Q Statistics of Actuator Fault 4n (EKF Type 4n, n/4p)

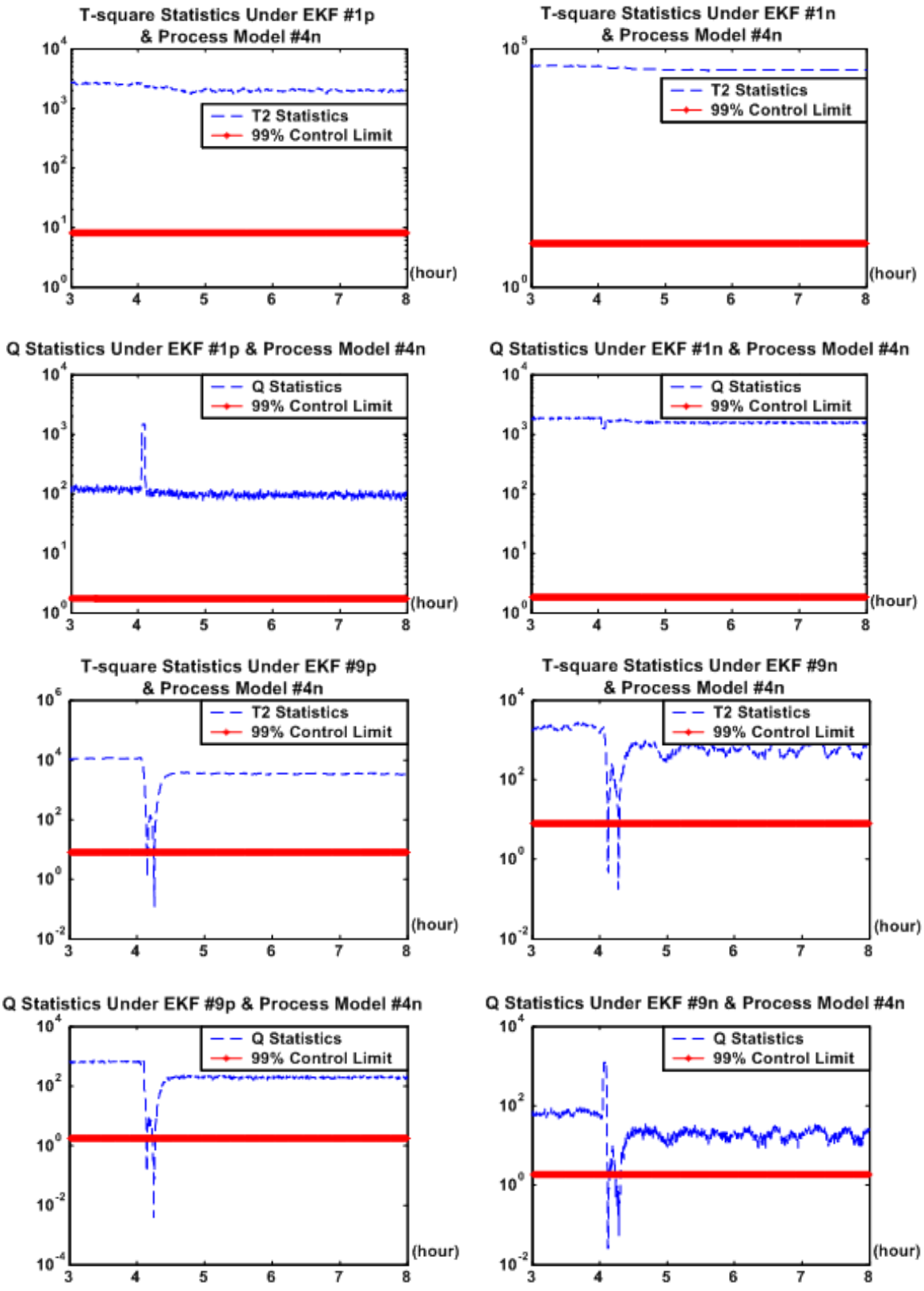

Figure 2.29: T2/Q Statistics of Actuator Fault 4n (EKF Type 1p/1n, 9p/9n)

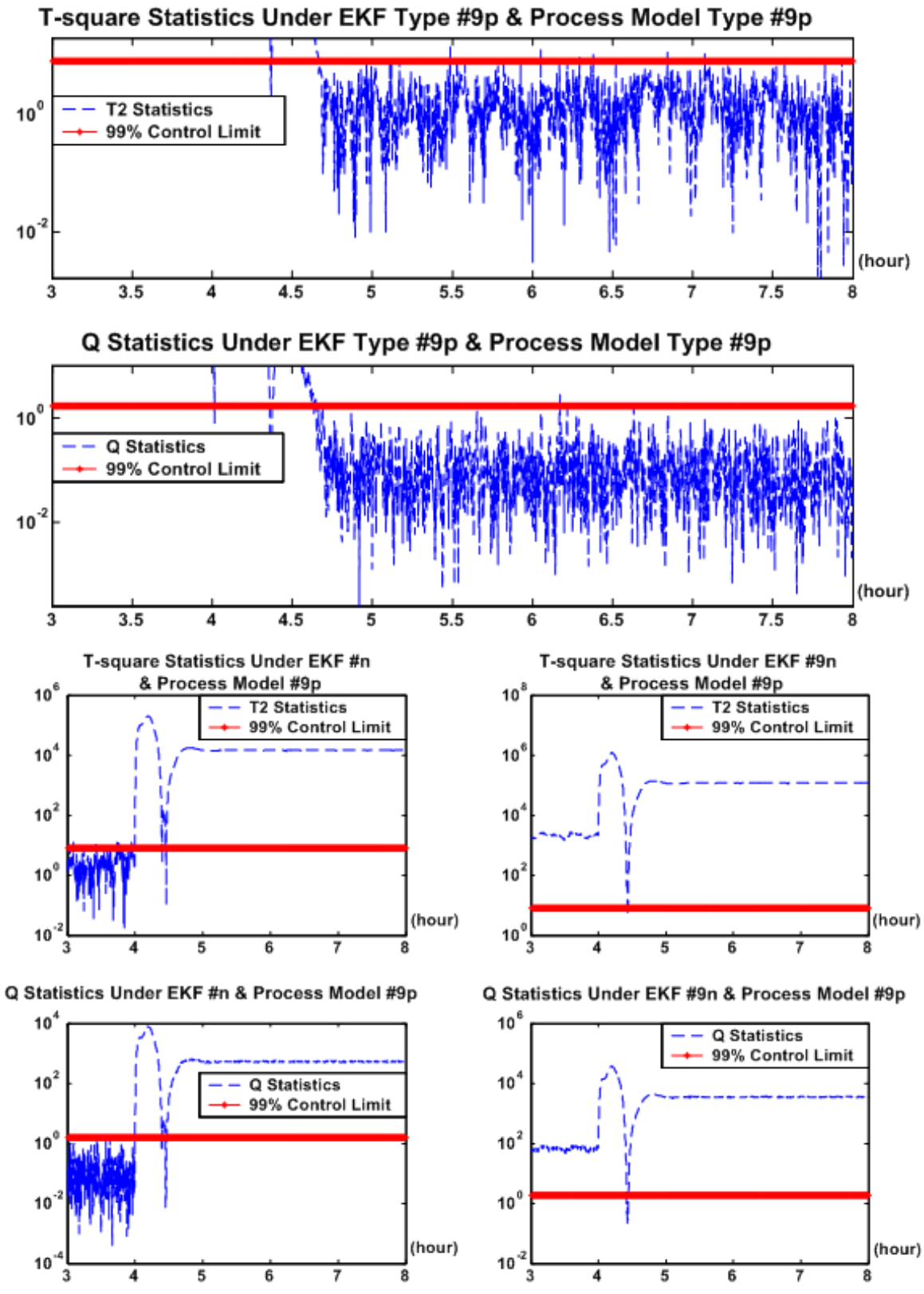

Figure 2.30: T2/Q Statistics of Process Fault 9p (EKF Type 9p, n/9n)

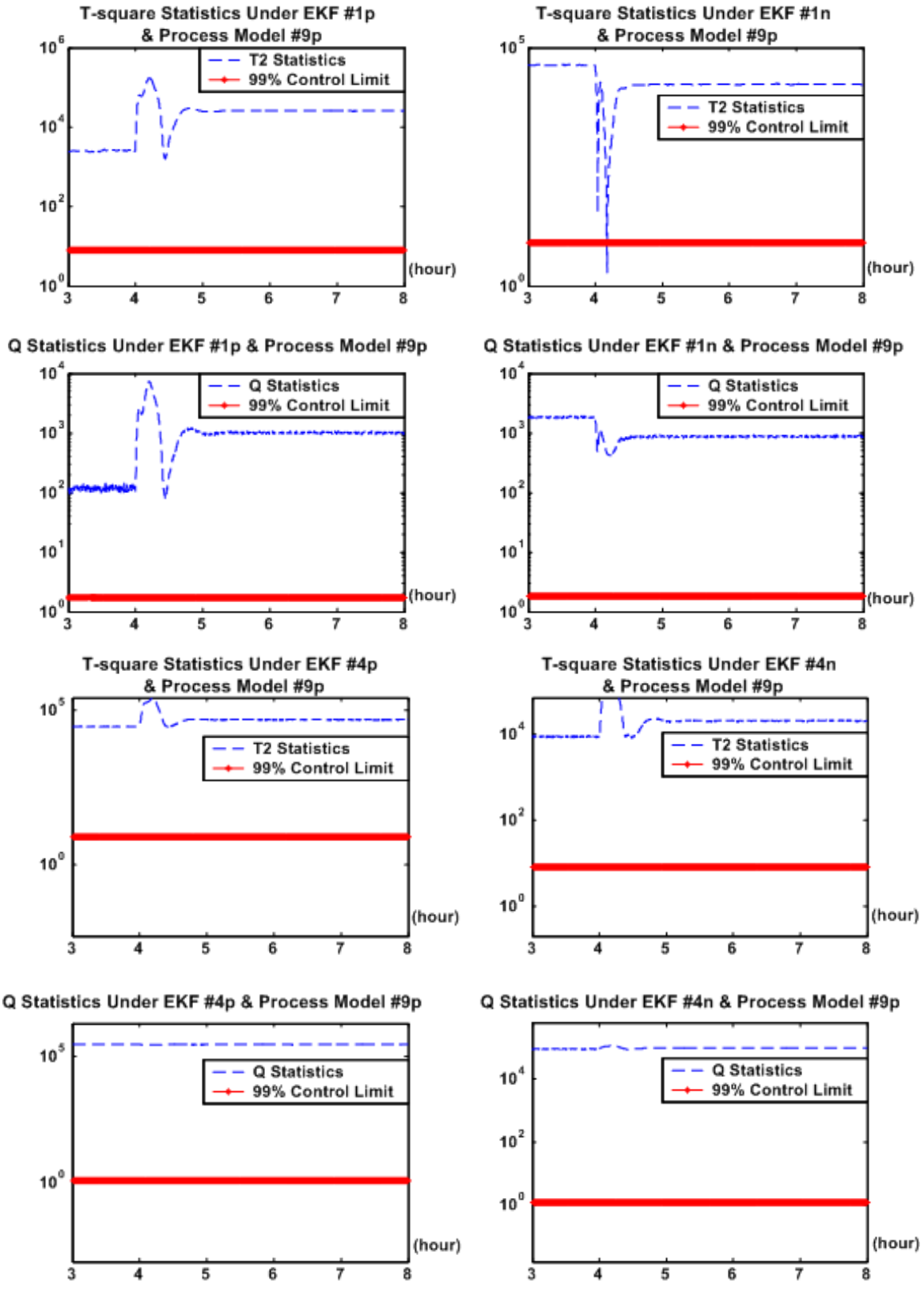

Figure 2.31: T2/Q Statistics of Process Fault 9p (EKF Type 1p/1n, 4p/4n)

3. **Process Under Actuator Fault (f4n)**

   The T^2 and Q statistics when the process is under actuator fault #4n and EKF is under normal condition and 6 fault conditions are given in Fig. 2.28 and 2.29.

4. **Process Under Process Fault (f9p)**

   The T^2 and Q statistics when the process is under process fault #9p and EKF is under normal condition and 6 fault conditions are given in Fig. 2.30 and 2.31.

2.4.3.3 PCA Fault Evaluation Test Results

As can be seen from the MMEKF simulation results, the T^2 and Q scores of the mismatched case are way beyond their control limits. This feature gives a very good fault detection result. Notice the MMEKF simulation results are based on the condition that the matching EKF fault magnitude is set equal to the corresponding process fault magnitude. What if these two fault magnitudes are not equal? Test results showed that the T^2 and Q scores tend to go beyond control limits when the matching EKF fault magnitude is not equal to the corresponding process fault magnitude. The minimum deviation of the matching EKF fault magnitude from the corresponding process fault, which makes the T^2 and Q scores violate their control limits, is different for different fault types. The sensitivity of the fault magnitude deviation is not a issue studied in the thesis because apparently it relates to the specific process and fault type. However, this interesting observation provides an idea of how to evaluate the faults as discussed in section 2.3.2.1.

Numerous tests show that the more the EKF fault magnitude deviates from the

process fault magnitude, the more the residuals deviate from zero-mean, in turn the larger the T^2 and Q scores. The minimum of the T^2 and Q scores can be found when the EKF fault magnitude is exactly equal to the process fault magnitude. It is natural to assume that when the EKF type matches the process fault type, the (T^2/Q vs EKF fault ratio) would be an unimodal function in the admissible fault ratio range. This assumption is verified with test results. Now the problem of fault evaluation becomes a finite range search problem. Several methods are available including exhaustive search, Dichotomous search, interval halving, Fibonacci, and Golden section [135]. In this slow chemical process, the EKF estimation is only reliable after the transients settle down. But the transients take at least 0.2 hours in our simulation under the best situation. This implies that the search effort for the optimum (matching) ratio should not require a large number of sequential steps. Therefore, the only appropriate search method for this particular problem would be an exhaustive search, since it is a simultaneous search method in which all the experiments are conducted before any judgment is made regarding the location of the optimum point. Although more efficient sequential search methods are available, the ability of simultaneous searching for the optimum point to avoid the excessive time delay resulted from the sequential search methods makes the exhaustive search the best option.

The exhaustive search method consists of evaluating the objective function, T^2 and Q scores in our case, at a predetermined number of equally spaced points, n, in the interval [f_{min} f_{max}], and reducing the interval of uncertainty using the assumption of unimodality. In general, if the objective function is evaluated at n equally spaced

points in the original interval of uncertainty of length $L_0 = f_{\max} - f_{\min}$, and if the optimum value of the function (among the n function values) turns out to be at point f_j, $1 \leq j \leq n$, the final interval of uncertainty is given by

$$L_n = f_{j+1} - f_{j-1} = \frac{2}{n+1} L_0 \tag{2.97}$$

The PCA fault evaluation test result of process fault #1p is shown in Fig. 2.32. The number of equally spaced points n is 18. The T^2 and Q scores of seven EKF types (normal plus 6 faulty conditions) after the initial transients settle down are calculated. The EKF normal condition data should correspond to only one point in the graph, and is shown as a straight line with data point marker '*'. It is easy to see from both the T^2 and Q statistics plots that the process fault type is #1p since only the EKF type #1p has data that goes below the 99% control limit, and its T^2 and Q statistics are unimodal functions. The fault ratio range of fault #1p is $[0.02, 0.8]$, and the actual fault ratio is located between data point 6 and 8. The final interval of uncertainty of the first search effort is $[0.225, 0.307]$, which contains the true fault ratio. The number of equally spaced points n can be increased if a more precise fault ratio is desired.

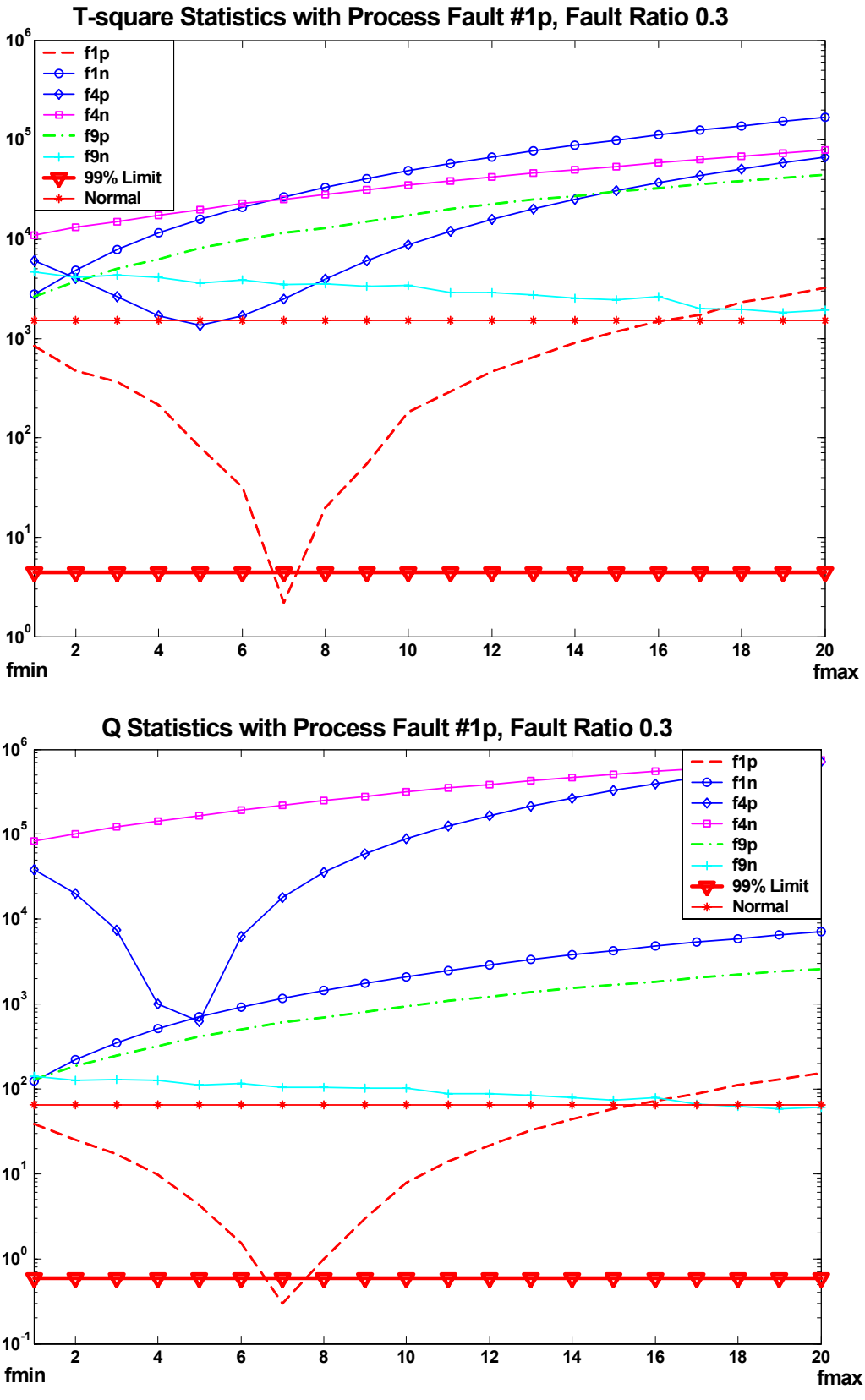

Figure 2.32: T2 and Q Statistics of Process Fault 1p (Fault Ratio 0.3)

Chapter 3

Data Driven Fault Diagnosis Approach: Principal Component Analysis

Model-based fault diagnosis uses mathematical models derived from physical principles and is based on parameter estimation or state estimation techniques. Unfortunately, comprehensive and robust models for complex processes are difficult, if not at all impossible, to develop and validate in both normal and fault modes. The modeling and fault diagnosis development is usually extremely time consuming and often the models are limited and can not characterize the process with all possible faults. Therefore, this approach is generally limited to those processes that are well understood or land themselves to a combination of physical/empirical modeling approaches.

During process operation, a large number of variables are routinely monitored and stored in computer databases. In some processes where first principle models describe the process is not well understood, or the process is too complicated, then measurement data is the only source for gaining a deeper understanding of the process. Various approaches utilizing measurement data have been proposed for fault diagnosis in process operations. Please refer to section 1.1.2 for a detailed literature survey

about data driven fault diagnosis approaches.

Process monitoring based on the multivariate statistical analysis of process data has been a very popular topic in recent years for industrial process control mainly because it is a model-free technique which is capable of handling a large number of variables (hundreds). These approaches use Principal Component Analysis (PCA) or Partial Least Squares (PLS). A basic review of PCA is shown in section 2.3.1.

Two fundamental statistical assumptions for the application of PCA are: measured variables are independent and identically distributed normal random variables; measured variables are uncorrelated in time (not auto-correlated). In practice, measurements from dynamic chemical processes do not satisfy these two assumptions, resulting in the loss of a statistical basis for the T^2 and Q statistics. It has been shown in [186] that the underlying distribution of data can vary substantially from normality without affecting the results, while auto-correlated measurements would definitely influence the results. Consequently, conventional PCA is not suitable for monitoring dynamic processes due to the presence of nonstationarities and time dependencies.

In recent years, research attention has been directed to these underlying assumptions and extensions of conventional PCA have been proposed to accommodate this issue. One important extension is combining wavelet analysis with PCA, taking advantage of the multi-scale capability of wavelets. Wavelet analysis is adopted to extract deterministic features and approximately de-correlate auto-correlated measurements in [3], [93], [192], or to filter out noise and rectify gross errors imbedded

in the measured data [174], [37]. Another important extension is dealing with the aspect of statistical distribution, using either distribution-free multivariate methods [71], or non-parametric empirical density estimates to describe the probability density function of measured data [20]. Two examples of these important extensions, namely Moving Principal Component Analysis (MPCA) and Multi-scale Principal Component Analysis (MSPCA), are presented in the following sections. The research effort on these extensions is beneficial to the data driven fault diagnosis on its own behalf, it is also helpful to resolve the problem for residual evaluation of MMEKF approach. Conventional PCA is good enough for residual evaluation of MMEKF, provided that the mathematical model used is a faithful replica of the process dynamics, and the true characteristics of fault and noise signals are known. But the uncertainty of the system structure, parameters and the effect of disturbances makes the residual deviate from whiteness substantially. A more robust residual evaluation scheme is needed when the model and disturbance uncertainty are inevitable. The extensions of conventional PCA discussed in this chapter are expected to improve the robustness of residual evaluation.

Three variations of conventional PCA, Adaptive PCA, Moving PCA, and Multi-scale PCA, are presented as three sections in this chapter respectively. The methodology and detailed process monitoring scheme are shown in each section. Finally, simulation studies and real industrial application complete this chapter.

3.1 Adaptive Principal Component Analysis

Industrial processes commonly demonstrate slow time-varying behaviors, such as catalyst deactivation, sensor and process drifting, equipment aging, and preventive maintenance and cleaning. The time-varying characteristics of industrial processes include: (i) changes in the mean; (ii) changes in the variance; and (iii) changes in the correlation structure among variables, including changes in the number of significant principal components. A major limitation of conventional PCA-based monitoring is that the PCA model, once built from the training data, is time-invariant, while most real industrial processes are time-varying [50]. When a time-invariant PCA model is used to monitor processes with the aforementioned normal changes, false alarms often result, which significantly compromise the reliability of the monitoring system.

An adaptive process monitoring approach is needed to accommodate to the time-varying process condition. Research effort has been made on the development of adaptive process monitoring recently. The need to monitor microelectronics manufacturing processes had been pointed out by Gallagher et al. [50]. Wold [189] discussed the use of exponentially weighted moving average (EWMA) filters in conjunction with PCA and PLS. A similar moving window scheme to identify significant modes in a simulated paper machine profile was proposed by Rigopoulos et al. [139]. Rannar et al., [133] used a hierarchical PCA for adaptive batch monitoring in a way that is similar to EWMA based PCA. A complete adaptation mechanism of recursive PLS was presented by Qin [125] and was followed by a similar recursive PCA algorithm [89].

3.1.1 Methodology

The adaptive PCA algorithm updates the PCA model recursively. It adapts for normal process change and this can help to reduce false alarms. A complete recursive PCA scheme should consider the following issues:

1. Recursive update of the mean for both covariance-based PCA and correlation-based PCA.

2. Efficient sample-wise update algorithms for the PCA calculation.

3. Recursive determination of the number of principal components, as the significant number of components changes.

4. Recursive determination of the confidence limits for T^2 and Q scores in real-time to facilitate adaptive monitoring.

The process variables are usually scaled to zero mean and unit variance in correlation-based PCA, changes in the statistics of the process variables can complicate the recursive computation. The recursive calculation for the correlation matrix with time varying mean and variance is shown below.

3.1.1.1 Recursive Update for the Correlation Matrix

In conventional PCA, the raw data matrix $X^0 \in \Re^{n \times p}$ of n samples (rows) and p variables (columns) is first normalized to a matrix X with zero mean and unit variance. Then the normalized matrix X is decomposed as follows:

$$\begin{aligned} X &= TP^T + \widetilde{X} \\ &= TP^T + \widetilde{T}\widetilde{P}^T = \begin{bmatrix} T & \widetilde{T} \end{bmatrix} \begin{bmatrix} P & \widetilde{P} \end{bmatrix}^T \end{aligned} \tag{3.1}$$

where $T \in \Re^{n \times q}$ and $P \in \Re^{p \times q}$ are the PC scores and loadings, respectively. The PCA projection reduces the original set of variables to q principal components. The residual matrix $\widetilde{X}$ can be further decomposed into $\widetilde{T}\widetilde{P}^T$ if desired. The decomposition is made such that $\begin{bmatrix} T & \widetilde{T} \end{bmatrix}$ is orthogonal and $\begin{bmatrix} P & \widetilde{P} \end{bmatrix}$ is orthonormal.

The correlation matrix of the variables can be calculated as $R = \frac{1}{n-1} X^T X$.

The columns of P are actually eigenvectors of R associated with the q largest eigenvalues, and the columns of $\widetilde{P}$ are the remaining eigenvectors, converting the calculation of the PCA model (P) to an eigenvector problem. Occasionally, the covariance matrix is used to derive a PCA model. In this case, the data is scaled to zero-mean, but the variance is unscaled. The variance scaling affects the relative weighting of all variables [67].

If a block of process data has been used to build an initial PCA model, we need to update the PCA model when a new block of data becomes available. Let $X_1^0 \in \Re^{n_1 \times p}$ be the raw initial data block. Then the mean of each column is given in the vector

$$\mu_1 = \frac{1}{n_1} \left(X_1^0\right)^T l_{n_1} \tag{3.2}$$

where $l_{n_1} = [1, 1, \ldots, 1]^T \in \Re^{n_1}$. The data scaled to zero mean and unit variance is given by

$$X_1 = \left(X_1^0 - l_{n_1}\mu_1^T\right)\Sigma_1^{-1} \tag{3.3}$$

where

$$\Sigma_1 = diag(\sigma_{1.1}, \ldots, \sigma_{1.p}) \tag{3.4}$$

whose i^{th} element is the standard deviation of the i^{th} sensor ($i = 1, \ldots, p$). The correlation matrix is

$$R_1 = \frac{1}{n_1 - 1} X_1^T X_1 \tag{3.5}$$

The new block of data is expected to augment the data matrix and used in the recursive calculation of the correlation matrix. Assume that μ_k, X_k, and R_k have been calculated when the k^{th} block of data is collected. The task for recursive calculation is to calculate μ_{k+1}, X_{k+1}, and R_{k+1} when the next block of data $X_{n_{k+1}}^0 \in \Re^{n_{k+1} \times p}$ is available. Denoting

$$X_{k+1}^0 = \begin{bmatrix} X_k^0 \\ \\ X_{n_{k+1}}^0 \end{bmatrix} \tag{3.6}$$

for all the $k+1$ blocks of data, the mean vector μ_{k+1} is related to μ_k by the following relation:

$$\left(\sum_{i=1}^{k+1} n_i\right)\mu_{k+1} = \left(\sum_{i=1}^{k} n_i\right)\mu_k + \left(X_{n_{k+1}}^0\right)^T l_{n_{k+1}} \tag{3.7}$$

Denoting $N_k = \left(\sum_{i=1}^{k} n_i\right)$, Eq. 3.7 yields the following recursive calculation:

$$\mu_{k+1} = \frac{N_k}{N_{k+1}}\mu_k + \frac{1}{N_{k+1}}\left(X^0_{n_{k+1}}\right)^T l_{n_{k+1}} \tag{3.8}$$

The recursive calculation of X_{k+1} is given by

$$\begin{aligned}
X_{k+1} &= \left[X^0_{k+1} - l_{k+1}\mu^T_{k+1}\right]\Sigma^{-1}_{k+1} \\
&= \left[\begin{bmatrix} X^0_k \\ X^0_{n_{k+1}} \end{bmatrix} - l_{k+1}\mu^T_{k+1}\right]\Sigma^{-1}_{k+1} \\
&= \begin{bmatrix} X^0_k - l_k\Delta\mu^T_{k+1} - l_k\mu^T_k \\ X^0_{n_{k+1}} - l_{n_{k+1}}\mu^T_{k+1} \end{bmatrix}\Sigma^{-1}_{k+1} \\
&= \begin{bmatrix} X_k\Sigma_k\Sigma^{-1}_{k+1} - l_k\Delta\mu^T_{k+1}\Sigma^{-1}_{k+1} \\ X_{n_{k+1}} \end{bmatrix}
\end{aligned} \tag{3.9}$$

where

$$\begin{aligned}
X_k &= \left(X^0_k - l_k\mu^T_k\right)\Sigma^{-1}_k \\
X_{n_{k+1}} &= \left(X^0_{n_{k+1}} - l_{n_{k+1}}\mu^T_{k+1}\right)\Sigma^{-1}_{k+1} \\
\Sigma_j &= diag\left(\sigma_{j.1}, \ldots, \sigma_{j.p}\right), j = k, k+1 \\
\Delta\mu_{k+1} &= \mu_{k+1} - \mu_k
\end{aligned} \tag{3.10}$$

Note that $l_k = [1, 1, \ldots, 1]^T \in \Re^{N_k}$.

The recursive computation of the standard deviation, derived in Appendix A.1, has the following relationship,

$$(N_{k+1}-1)\sigma^2_{k+1.i} = (N_k-1)\sigma^2_{k.i} + N_k\Delta\mu^2_{k+1}(i) + \left\| X^0_{n_{k+1}}(:,i) - l_{n_{k+1}}\mu_{k+1}(i) \right\|^2 \quad (3.11)$$

where $X^0_{n_{k+1}}(:,i)$ is the i^{th} column of the associated matrix; $\mu_{k+1}(i)$ and $\Delta\mu_{k+1}(i)$ are the i^{th} elements of the associated vectors.

Similarly, the recursive calculation of the correlation matrix, derived in Appendix A.2, has the following form,

$$\begin{aligned} R_{k+1} &= \frac{1}{N_{k+1}-1} X^T_{k+1} X_{k+1} \\ &= \frac{N_k-1}{N_{k+1}-1}\Sigma^{-1}_{k+1}\Sigma_k R_k \Sigma_k \Sigma^{-1}_{k+1} + \frac{N_k}{N_{k+1}-1}\Sigma^{-1}_{k+1}\Delta\mu_{k+1}\Delta\mu^T_{k+1}\Sigma^{-1}_{k+1} \\ &\quad + \frac{1}{N_{k+1}-1} X^T_{n_{k+1}} X_{n_{k+1}} \end{aligned} \quad (3.12)$$

Regarding the above recursive relation, note that:

1. The effect of mean changes $\Delta\mu_{k+1}$ on the correlation matrix is only a rank-one modification.

2. As a special case, if the model needs to be updated after every new sample is taken, the recursive relation reduces to:

$$R_{k+1} = \frac{k-1}{k}\Sigma^{-1}_{k+1}\Sigma_k R_k \Sigma_k \Sigma^{-1}_{k+1} + \Sigma^{-1}_{k+1}\Delta\mu_{k+1}\Delta\mu^T_{k+1}\Sigma^{-1}_{k+1} + \frac{1}{k} x_{k+1} x^T_{k+1} \quad (3.13)$$

   which is two rank-one modifications.

3. Often old data are exponentially ignored as they do not represent the current process. The recursive calculations for Eqs. 3.8, 3.11 and 3.12 with a forgetting

factor β are:

$$\mu_{k+1} = \beta\mu_k + (1-\beta)\frac{1}{n_{k+1}}\left(X^0_{n_{k+1}}\right)^T l_{n_{k+1}} \tag{3.14}$$

$$\sigma^2_{k+1.i} = \beta(\sigma^2_{k.i} + \Delta\mu^2_{k+1}(i)) + (1-\beta)\frac{1}{n_{k+1}}\left\|X^0_{n_{k+1}}(:,i) - l_{n_{k+1}}\mu_{k+1}(i)\right\|^2 \tag{3.15}$$

and

$$R_{k+1} = \beta\Sigma^{-1}_{k+1}(\Sigma_k R_k \Sigma_k + \Delta\mu_{k+1}\Delta\mu^T_{k+1})\Sigma^{-1}_{k+1} + (1-\beta)\frac{1}{n_{k+1}}X^T_{n_{k+1}}X_{n_{k+1}} \tag{3.16}$$

for $N_k \gg 1$. In the above relations, $0 < \beta \leq \frac{N_k}{N_{k+1}} < 1$ is the forgetting factor. Smaller β tends to forget old data more quickly, whereas

$$\beta = \frac{N_k}{N_{k+1}} \tag{3.17}$$

recovers the case of no forgetting. Similar to the window size in a moving window approach, the forgetting factor is a tuning parameter that varies depending on how fast the normal process can change.

3.1.1.2 Recursive Determination of the Number of PCs

Since the number of significant principal components can change over time, it is necessary to determine this number recursively in adaptive PCA modelling. There are many ways of determining the number of PCs in batch-wise PCA, including:

- Cross-validation [188], [116]
- Cumulative percent variance [98]
- Average eigenvalues

- Imbedded error function [98]
- Xu and Kailath's approach [191]
- Akaike information criterion [1], [178]
- Minimum description length criterion [178], [140]
- Scree test [15]
- Variance of reconstruction error [38]

However, not all the approaches are suitable for recursive PCA. For example, the cross-validation approach is not suitable because old data are not representative for the current process. Therefore, we consider only the following methods to determine the number of PCs recursively.

1. Cumulative percent variance (CPV): The CPV is a measure of the percent variance captured by the first l_k PCs:

$$CPV(l_k) = \frac{\Sigma_{j=1}^{l_k} \lambda_j}{\Sigma_{j=1}^{p} \lambda_j} 100\%$$

Since only the first l_k eigenvalues are calculated in the block-wise adaptive PCA, we use the following formula:

$$CPV(l_k) = \frac{\Sigma_{j=1}^{l_k} \lambda_j}{trace(R_k)} 100\% \tag{3.18}$$

The number of PCs is chosen when the CPV reaches a predetermined limit, say 95%. This is one of the most popular methods to determine the number of PCs and is used in this thesis for the correlation-based PCA.

2. Average eigenvalue (AE): The AE approach selects the eigenvalues greater than the mean of all eigenvalues as the principal eigenvalues and discards those smaller than the mean. Therefore, for R_k, only the eigenvalues greater than $\frac{trace(R_k)}{p}$ are used as the principal eigenvalues.

3. Imbedded error function (IEF). The IEF is given by:

$$IEF(l_k) = \sqrt{\frac{l_k \theta_1}{N_k p(p - l_k)}} \tag{3.19}$$

where

$$\theta_1 = \sum_{j=l_k+1}^{p} \lambda_j = trace(R_k) - \sum_{j=1}^{l_k} \lambda_j \tag{3.20}$$

and N_k is the length of the data set used to compute R_k. If the forgetting factor β is employed, N_k is estimated from Eq. 3.17 as $N_k = \frac{n_k}{1-\beta}$.

If the noise in each variable has the same variance, IEF(l_k) will have a minimum corresponding to the number of PCs. Note that IEF works for covariance-based PCA only.

4. Xu and Kailath's approach: Xu and Kailath [191] determine the number of PCs for a covariance matrix based on a few of the largest eigenvalues. This approach also assumes that the smallest eigenvalues of the covariance matrix R_k are equal. If the remaining eigenvalues after l_k are identical,

$$\phi_{l_k} = \log \left(\frac{\sqrt{\frac{\theta_2}{p-l_k}}}{\frac{\theta_1}{p-l_k}} \right)$$

should be approximately zero, where

$$\theta_2 = \sum_{j=l_k+1}^{p} \lambda_j^2 = trace(R_k^2) - \sum_{j=1}^{l_k} \lambda_j^2 \tag{3.21}$$

Xu and Kailath [191] showed that if the observation vector x_k is sampled from a zero-mean multivariate normal distribution, then $N_k(p-l_k)\phi_{l_k}$ will asymptotically have a chi-squared distribution with $\frac{1}{2}(p-l_k)(p-l_k+1)-1$ degrees of freedom. Therefore, with a chosen significance level α, we can select the threshold χ^2_α for ϕ_l. If $N_k(p-l_k)\phi_l \leq \chi^2_\vartheta$, we accept the null hypothesis that $l = l_k$. This approach is valid only for covariance-based PCA.

5. Information theoretic criteria: Wax and Kailath [178] proposed to determine the number of PCs for a covariance matrix based on the AIC [1] and MDL [140] criteria. This approach assumes that the observation vector x_k is sampled from a zero-mean multivariate normal distribution and the covariance matrix has repeated smallest eigenvalues. Wax and Kailath [178] proposed a practical criterion and showed that the following two functions will be minimized when l_k is equal to the number of PCs.

$$AIC(l_k) = -2\log\left(\frac{\prod_{j=l_k+1}^{p}\lambda_j^{\frac{1}{p-l_k}}}{\frac{\theta_1}{p-l_k}}\right)^{N_k(p-l_k)} + 2M$$

$$MDL(l_k) = -2\log\left(\frac{\prod_{j=l_k+1}^{p}\lambda_j^{\frac{1}{p-l_k}}}{\frac{\theta_1}{p-l_k}}\right)^{N_k(p-l_k)} + M\log(N_k)$$

where $\prod_{j=l_k+1}^{p}\lambda_j = \frac{\det(R_k)}{\prod_{j=1}^{l_k}\lambda_j}$

For real valued signals, $M = \frac{l_k(2m-l_k+1)}{2}$, and for complex valued signals, $M = l_k(2m - l_k)$.

Wax and Kailath point out that MDL gives a consistent estimate but AIC usually overestimates the number of PCs when N_k is very large. In addition, the AIC and MDL criteria are valid only for covariance-based PCA and fail for rank deficient covariance matrices, e.g. more variables than samples.

6. Variance of reconstruction error (VRE): Dunia and Qin [38] propose selecting the number of PCs based on the best reconstruction of the process variables. An important feature of this approach is that the index has a minimum corresponding to the best reconstruction and is valid for both covariance and correlation-based PCA, no matter how the observation vector x_k is distributed. The VRE method assumes that each sensor (or group of sensors) is faulty and reconstructed from other sensors based on the PCA loadings, P_{l_k}. The variance of the reconstruction error for the i^{th} sensor is [39]:

$$v_i = \frac{r_k(i,i) - 2c_k^T(:,i)r_k(:,i) + c_k^T(:,i)R_k c_k(:,i)}{(1 - c_k(i,i))^2}, i = 1, \ldots, p \tag{3.22}$$

where $c_k(:,i)$ and $c_k(i,i)$ are the i^{th} column and ii^{th} element of matrix $P_{l_k}P_{l_k}^T$, respectively. Similarly, $r_k(:,i)$ and $r_k(i,i)$ are the i^{th} column and ii^{th} element of matrix R_k, respectively. The VRE is defined to account for all sensors,

$$VRE(l_k) = \sum_{i=1}^{p} \frac{v_i}{var(x_i)} \tag{3.23}$$

where $var(x_i)$ stands for the variance of the i^{th} element of the observation vector. Therefore, $VRE(l_k)$ can be calculated recursively using only the eigenvalues and

eigenvectors of R_k up to l_k. The VRE method selects the number of PCs that gives a minimum VRE.

3.1.1.3 Adaptation of the Confidence Limits for T^2 and Q scores

For slowly time-varying processes, the confidence limits for T^2 and Q scores will change with time, making adaptation of these limits necessary for on-line monitoring.

The upper limit for the Q statistic is given by Eqs. 2.69, 2.71, where z_α is the normal deviate corresponding to the upper $1-\alpha$ percentile. In the recursive implementation, θ_1, θ_2, θ_3 and h_0 are updated using the l_k largest eigenvalues after each new data block, making the limit C_α, time-varying.

The control limit for the T^2 statistic is given by Eq. 2.67 with p replaced by l_k or follows a chi-squared distribution with l_k degrees of freedom and significance level α, $\chi^2_\alpha(l_k)$. Since l_k can be time-varying, the control limit for the T^2 statistic also needs to be updated for each new data block.

3.1.2 Process Monitoring Scheme

An adaptive real time monitoring scheme using the proposed adaptive PCA algorithm is described in this section, which addresses the issues of missing data and outliers that frequently occurs in industrial processes with a large number of sensors.

3.1.2.1 Missing Data Treatment

Missing measurements often occur when sensors fail or are taken off-line for routine maintenance. In other situations, measurements are removed from a data set because gross measurement errors occur or samples are simply not collected at the required time. In these cases, the measurement data are missing at random times. In other situations, missing data occurs on a regular basis. An example is when sensors have different sampling frequencies. The standard procedure for handling missing data in PCA model building is based on the NIPALS algorithm which computes one vector at a time [100]. Once a PCA model has been built, three approaches are proposed by Nelson et al. [108] to treat missing data. First method is called single component projection and is derived from the NIPALS algorithm but is a non-iterative approach. The second method handles missing data in PCA by projection to the model plane. The third method replaces missing data with the conditional mean (expected values from the conditional normal distribution given the present data and the current estimate of the mean and covariance). Tests using simulation and industrial data sets show that in general, all three methods perform reasonably well with moderate amounts of missing data (up to 20% of the measurements). In extreme cases where critical combinations of measurements are missing, the conditional mean replacement method is generally superior to other approaches and is adopted in this work.

3.1.2.2 Outlier Replacement

The data obtained from any process may contain outliers (data either are irrelevant, grossly erroneous or abnormal in some other way, compared with the majority of the data) as a result of process characteristics, faulty sensors, equipment failures, transient effects, or during the transference of values acquired by analog/digital converters [60]. These outliers may lead to incorrect conclusions, if the data are analyzed without accounting for their effects. Hence, it is important that these erroneous observations be given less importance.

Heuristically, observations that exceed five standard deviations are considered to be outliers. Outliers in a univariate data set can be easily identified by visual inspection, however, as the dimension of the system increases, this task becomes cumbersome and, sometimes, impossible. In real-time applications, if one ignores or simply deletes the outlying observations by heuristic reasoning, the process operating performance might be adversely affected. Hence, a direct and efficient outlier replacement methodology that performs in an unsupervised manner is required. Among a wide range of robust location estimators, the M-estimator is the 'most robust' one since it has minimum efficiency loss when the outliers are sampled from various distributions [60]. Therefore, the simple, yet powerful moving median (MM) filter [36] is selected for real-time outlier replacement. In this nonlinear signal processing technique, the median of a window containing an odd number of observations is found by sliding it over the entire one-dimensional signal. As a result, the original signal is freed from outliers. Unlike the moving average (MA) filter, the MM filter does not have the

blurring effect on signals.

Another multivariate outlier replacement approach is proposed by Hoo et al. [63]. The objective was to find an accurate estimate of the covariance/correlation matrix of the data so that a PCA model might be developed that could then be used for monitoring and fault detection and identification. The methodology produces robust estimates of the mean (location) and standard deviation (scale) iteratively, yielding a robust set of data. The robust multivariate outlier replacement is performed in the score space generated by PCA, which filters the outliers in the multivariate instead of univariate sense. Hampel's and Huber's M-estimators are used to replace the outliers with values more representative of the true process normal behavior. One of the key issues here is obtaining the estimates of the true mean and standard deviation of the normal data for PCA when the data contain outliers. The median is used to estimate the mean and the MADM (median absolute deviation of the median) to estimate the standard deviation.

3.1.2.3 A Complete Adaptive Monitoring Scheme

Let $C_{k,\alpha}$ and $F_\alpha(l_k)$ denote the confidence limits of Q and T^2 statistics, and P_{l_k} the PCA loadings at time t_k, respectively. With the knowledge of $C_{k,\alpha}$, $F_\alpha(l_k)$, and P_{l_k}, the adaptive process monitoring, including normal data adaptation and fault detection, can be implemented in real-time, as follows:

1. Initialization stage: Build an initial reference PCA model with a certain number of samples, say, at least 50. Calculate the T^2 and Q scores of these samples and

replace those samples which exceed control limits using the outlier replacement approaches mentioned above. Make sure no sample goes outside the control limits. Let t_0 denote the time of the initial stage, then calculate the number of PCs l_0, P_{l_0}, $C_{0,\alpha}$, and $F_\alpha(l_0)$ and starts the iteration process.

2. Collect new data sample(s) at time t_k. If there are missing values in the samples, replace them using the conditional mean replacement method [108]. Preprocess new sample(s) using the robust outlier replacement method or the moving median filter.

3. Calculate T^2 and Q scores of new sample(s). If $T^2 > F_\alpha(l_k)$ or $Q > C_{k,\alpha}$, a possible process fault alarm is triggered and recorded. If a number of consecutive samples trigger the alarm (or the percentage of recent samples that exceed the control limits is above a certain threshold), the model update is terminated and a process fault is declared.

4. Update the PCA model based on the new data block if no process alarms have occurred. Calculate l_{k+1}, $P_{l_{k+1}}$, $C_{k+1,\alpha}$, and $F_\alpha(l_{k+1})$. Set $k = k + 1$ and go to step 2.

3.2 Moving Principal Component Analysis

Despite the success of conventional PCA in monitoring multivariate process, one of its limitations is that it cannot detect the change of correlation among process variables as long as both T^2 and Q statistics are inside the control limits. A statis-

tical process monitoring method is proposed to address this limitation. The method is based on the idea that a change of operating condition, i.e. a change of correlation among process variables, can be detected by monitoring directions of principal components. To calculate principal components successively, a time-window is introduced. Because PCA is applied on-line by moving the time-window, this method is termed moving principal component analysis (MPCA).

3.2.1 Index for Monitoring

In order to detect a change of the PCs, reference PCs representing the normal operating condition are defined, and the differences between the reference PCs and PCs representing the current operating condition are used as indices for monitoring. The following index A_i is proposed for evaluating the change of the i^{th} PC,

$$A_i(k) = 1 - \left| w_i(k)^T w_{i0} \right| \tag{3.24}$$

where $w_i(k)$ denotes the i^{th} PC calculated at time t_k, and w_{i0} denotes the i^{th} reference PC. Both $w_i(k)$ and ω_{i0} are unit vectors. The index A_i is based on the inner product of the PCs, i.e. the angle of the PCs. When the i^{th} PC representing a current operating condition is equivalent to its reference, A_i becomes zero. However, A_i becomes one when $w_i(k)$ is orthogonal to w_{i0}.

MPCA can detect a change of correlation among process variables, which is difficult to detect by conventional PCA (CPCA) with T^2 and Q statistics. Another nice feature of MPCA is that the MPCA indices do not assume any probability distribution. It is a distribution-free multivariate method. However, the index A_i does

not function well when variances of several PCs are similar to each other, because in such a situation the directions of PCs might change abruptly while the correlation among process variables is unchanged. To cope with this problem, a change of subspace spanned by several PCs with similar variances should be monitored instead of a change of each PC.

The change of subspace, when an p-dimensional subspace F_1 changes to F_2, is defined as the maximum distance between an arbitrary unit vector x in F_2 and the subspace F_1. When $v_1, v_2, \ldots, v_p$ and $w_1, w_2, \ldots, w_p$ are n-dimensional orthonormal bases of F_1 and F_2, respectively, the orthogonal projection of a unit vector x in F_2 on F_1 is given by Px, where

$$P = VV^T \tag{3.25}$$

$$V = [v_1, v_2, \ldots, v_p] \tag{3.26}$$

The change of subspace is defined as the maximum of $\|x - Px\|$ and the x that maximizes $\|x - Px\|$ must be determined. This is equivalent to determining the x that minimizes $\|Px\|$. Since x is a unit vector in F_2, it can be represented as

$$x = Wy \tag{3.27}$$

$$W = [w_1, w_2, \ldots, w_p] \tag{3.28}$$

where y is an p-dimensional unit vector. Then, the problem can be rewritten as follows:

$$\min x^T Px = y^T W^T PWy \tag{3.29}$$

s.t. $\|y\| = 1$.

Let $L(y) = y^T W^T PWy - \lambda y^T y$, then the necessary conditions can be derived as follows:

$$W^T PWy - \lambda y = 0 \tag{3.30}$$

where λ is a Lagrange multiplier. This condition indicates that the vector y to be determined is an eigenvector of the matrix $W^T PW$ and that λ is the corresponding eigenvalue. In addition, the equation

$$y^T W^T PWy = y^T \lambda y = \lambda \tag{3.31}$$

indicates that the minimum of $y^T W^T PWy$ is equivalent to the minimum eigenvalue of the matrix $W^T PW$, that is, the smallest λ.

As a result, the unit vector y must be the eigenvector corresponding to the minimum eigenvalue λ_{min} of the matrix $W^T PW$, and the change of subspace is given by:

$$A_{1-m} = \|Wy - PWy\| = \sqrt{1 - \lambda_{min}} \tag{3.32}$$

where A_{1-m} is the index for monitoring the change of the subspace spanned by the first m PCs. The index A_{1-m} enables the proposed method to function successfully even when variances of several PCs are similar to each other.

3.2.2 Process Monitoring Scheme

To apply the proposed monitoring method, reference PCs and control limits must be determined. Because MPCA is a distribution-free multivariate method, its control limits have to be determined empirically. The algorithm is described as follows:

1. Get measured data when the process is operating under the normal condition. Normalize each column (variable) of the data matrix, i.e. adjust it to a zero mean and unit variance random vector.

2. Apply PCA to the data matrix, and define a reference PC, w_{i0}.

3. Determine the size (steps) of the moving window, w. Generate data sets with w samples from the data by moving the time window. Apply PCA to the data sets, and calculate PCs, w_i.

4. Calculate the index A_i or A_{1-m}, and determine the control limits.

5. For on-line monitoring, the data matrix representing the current operating condition is updated by moving the time-window step by step, and it is scaled with the mean and the variance obtained at step 1. Then, PCA is applied to the $w \times p$ (p is the number of variables) data matrix, and the index A_i or A_{1-m} is calculated at each step. If the index is outside the control limit, the process is considered to be out of control.

If moving the time window and updating the PCs are computationally expensive, a sliding window PCA algorithm can be used to update the PCs recursively (see Appendix B).

3.3 Multi-Scale Principal Component Analysis

Conventional PCA methods operate at a fixed scale, the scale of the sampling frequency, and are best for detecting change at a single scale. However, data from most practical processes are inherently multiscale in nature due to:

1. events occurring at different locations and with different localizations in time and frequency.

2. non-stationary stochastic processes whose energy or power spectrum changes with time and/or frequency.

3. variables are measured at different sampling rates or there is missing data.

For instance, components of measurement noise are mainly located in the high frequency bands, while basic process dynamics are mainly localized in low frequency bands. Therefore, a multiscale technique like wavelets becomes a natural choice to separate these two quantities.

Many existing Statistical Process Control (SPC) methods assume uncorrelated measurements, whereas, in practice, auto-correlated measurements are extremely common. A common approach for SPC of auto-correlated measurements is to de-correlate them by fitting a time series model and monitoring the residual error. However, this approach is often not practical for industrial use, particularly for multivariate processes, due to their high dimensionality. For multivariate SPC, the dynamics are captured by a linear time series model using methods such as Dynamic PCA

(DPCA) [83] or subspace identification [107]. However, the autocorrelation is often still present in the measurements. Traditional T^2 and Q control charts are then applied in the space of the selected latent variables and their residuals. Nevertheless, these approaches often work better than static PCA for auto-correlated data.

In recent years, wavelets have become popular for analyzing multiscale or auto-correlated measurements due to their ability to compress multiscale features and approximately de-correlate many auto-correlated stochastic processes [35]. Thus, wavelet coefficients provide compact information about a signal at different localizations in both time and frequency. Also, the wavelet coefficients of many stochastic processes are approximately uncorrelated since the wavelets are approximate eigenfunctions of many mathematical operators [12]. Wavelet based multiscale methods have been used to develop for improved solutions of many problems including data compression, estimation, feature extraction, filtering, and change detection [35], [99], [113], [144].

To overcome the above mentioned limitations of conventional PCA, an approach called Multi-Scale Principal Component Analysis (MSPCA) is developed in [3], by combining the ability of PCA to extract the relationship between the variables and to reduce the cross-correlation between variables with the ability of wavelets to extract features in the measurements and approximately de-correlate the autocorrelation within a given measurement signal. Theoretical analysis and application to industrial data in [2] indicate that MSPCA is a good general method for SPC of processes containing features of different types and sizes and at different scales, from

uncorrelated or auto-correlated measurements.

3.3.1 Wavelets

Wavelets are a family of basis functions that are localized in both time and frequency, and may be represented as

$$\psi_{su}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s}) \tag{3.33}$$

where s and u represent the dilation and translation parameters, respectively. It is common to discretize these parameters dyadically as, $s = 2^m$, $u = 2^m k$, $(m, k) \in Z^2$, and the family of wavelets is represented as

$$\psi_{mk}(t) = 2^{-m/2}\psi(2^{-m}t - k) \tag{3.34}$$

where $\psi(t)$ is the mother wavelet, m and k are the dilation and translation parameters, respectively. The translation parameter determines the location of the wavelet in the time domain, while the dilation parameter determines the location in the frequency domain as well as the scale or extent of the time-frequency localization. These choices of s and u allow the wavelets to be orthonormal [29], but cause a downsampling (dyadic reduction or decimation) in the number of wavelet coefficients from one scale to the next.

Any signal can be decomposed into its contributions from multiple scales as a weighted sum of dyadically discretized orthonormal basis functions:

$$x(t) = \sum_{m=1}^{L}\sum_{k=1}^{N} d_{mk}\psi_{mk}(t) + \sum_{k=1}^{N} a_{Lk}\phi_{Lk}(t) \tag{3.35}$$

where x is a time signal, d_{mk} represents the wavelet or detail signal coefficient at scale m and location k, and a_{Lk} represent the scaled signal or scaling function coefficient of $\phi_{Lk}(t)$ at the coarsest scale L and location k. The scaling function or father wavelet, $\phi_{mk}(t)$, captures the low frequency content of the original signal that is not captured by the wavelets at the finer scales. An example of wavelet decomposition of the signal $x(t) = \begin{cases} w(t), & 0 \leq t < 100 \\ w(t) + 3, & 100 \leq t \leq 150 \\ w(t), & t > 150 \end{cases}$ is shown in Fig. 3.1, where $w(t)$ is a white noise with zero mean and unit variance. In this case, the signal is decomposed to a coarsest scale represented by $L = 3$. This decomposition is based on the Haar wavelet and corresponding scaling function which are defined as

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2} \\ -1, & \frac{1}{2} \leq t < 1 \\ 0, & otherwise \end{cases} \tag{3.36}$$

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & otherwise \end{cases} \tag{3.37}$$

The lowest frequency content of the signal is represented on a set of scaling functions, as depicted in Fig. 3.1(e). The number of wavelet and scaling function coefficients decreases dyadically at coarser scales due to the dyadic discretization of the dilation and translation parameters. Efficient methods have been developed for decomposing a signal given a family of wavelet basis functions based on convolution with a corresponding set of filters [99]. Convolution with a low pass filter, H, represents projection on the scaling function, and convolution with a high pass filter, G,

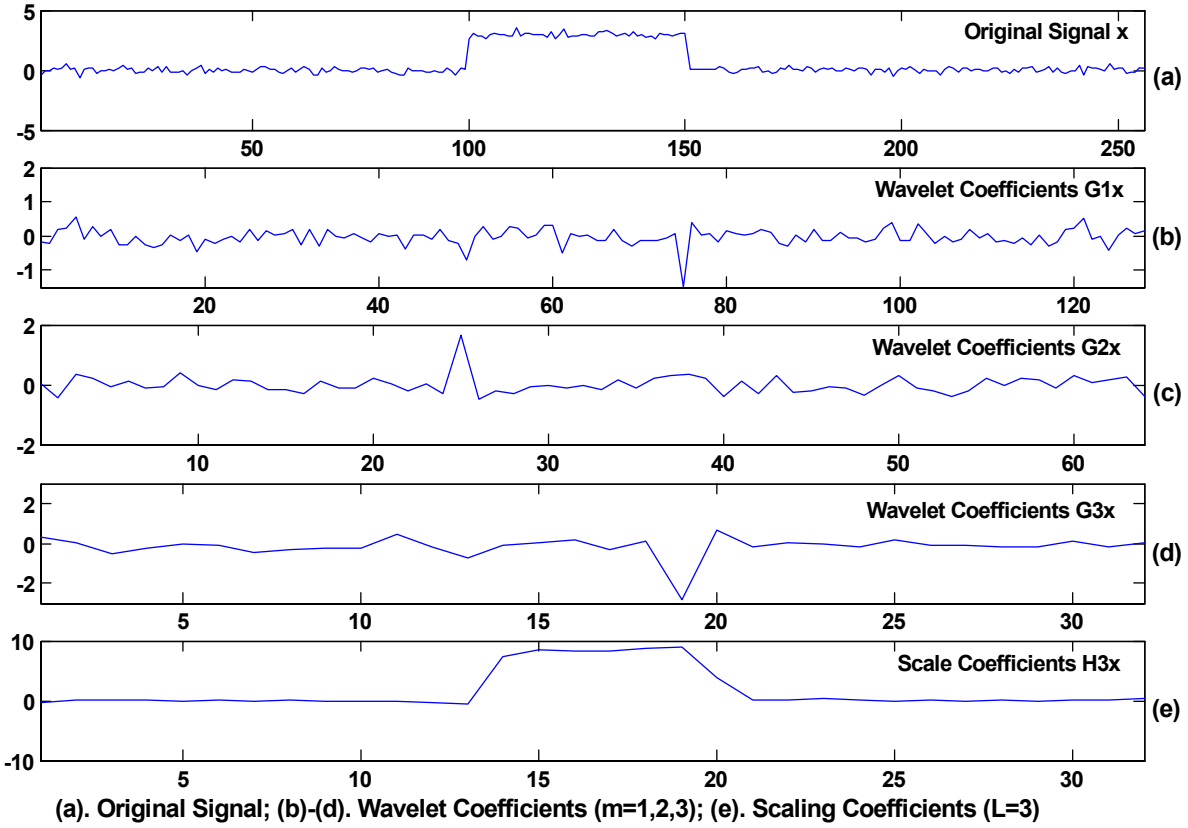

Figure 3.1: Wavelet Decomposition and Separation of Stochastic and Deterministic Components

represents projection on a wavelet. For example, the filter coefficients for Haar scaling functions and wavelets are $H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ and $G = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$ respectively. Thus, the coefficients at different scales may be obtained as

$$a_m = Ha_{m-1}, \quad d_m = Ga_{m-1} \tag{3.38}$$

where d_m is the vector of wavelet coefficients at scale m, and a_m is the vector of scaling function coefficients. The original data are considered to be the scaling function coefficients at the finest scale, that is, $x = a_0$. Eq. 3.38 can also be represented in terms of the original measured data vector, x, as

$$a_m = H_m x, \quad d_m = G_m x \tag{3.39}$$

where H_m is obtained by applying the H filter m times, and G_m is obtained by applying the H filter $(m-1)$ times and the G filter once. The original data may be reconstructed exactly from its wavelet coefficients at all scales, d_m for $m = 1, 2, ..., L$, and scaling function coefficients at the coarsest scale, a_L.

One useful property of wavelets is that although they are not known to be the exact eigenfunctions or principal components of any operators, they are the approximate eigenfunctions of a large variety of operators [32]. Consequently, the wavelet coefficients of most stochastic processes are approximately de-correlated.

The variance of the wavelet coefficients at different scales represents the energy of the stochastic process in the corresponding range of frequencies, and corresponds to its power spectrum. Thus, for an uncorrelated Gaussian stochastic process or white noise, the variance of the wavelet coefficients is constant at all scales, whereas for colored noise, the variance decreases at finer scales.

The ability of wavelets to de-correlate the data is depicted in Fig. 3.2. The original signal, $x_k = 0.9x_{k-1} + 0.5w_k$, is from a simulated stationary stochastic process, where w_k is a white noise with zero mean and unit variance. Its autocorrelation function is shown in Fig. 3.2(a). The wavelet coefficients and corresponding autocorrelation functions shown in Fig. 3.2(b), (c), and (d) indicate that the coefficients are approximately uncorrelated. The last scaled signal in Fig. 3.2(e) contains residual correlation, which may be further reduced by decomposing to coarser scales. The variance of the wavelet coefficients at each scale varies according to the power spectrum of the original signal. Thus, the variance of the wavelet coefficients and the last

scaled signal in Fig. 3.2(b), (c), (d), (e), increases with decreasing frequency.

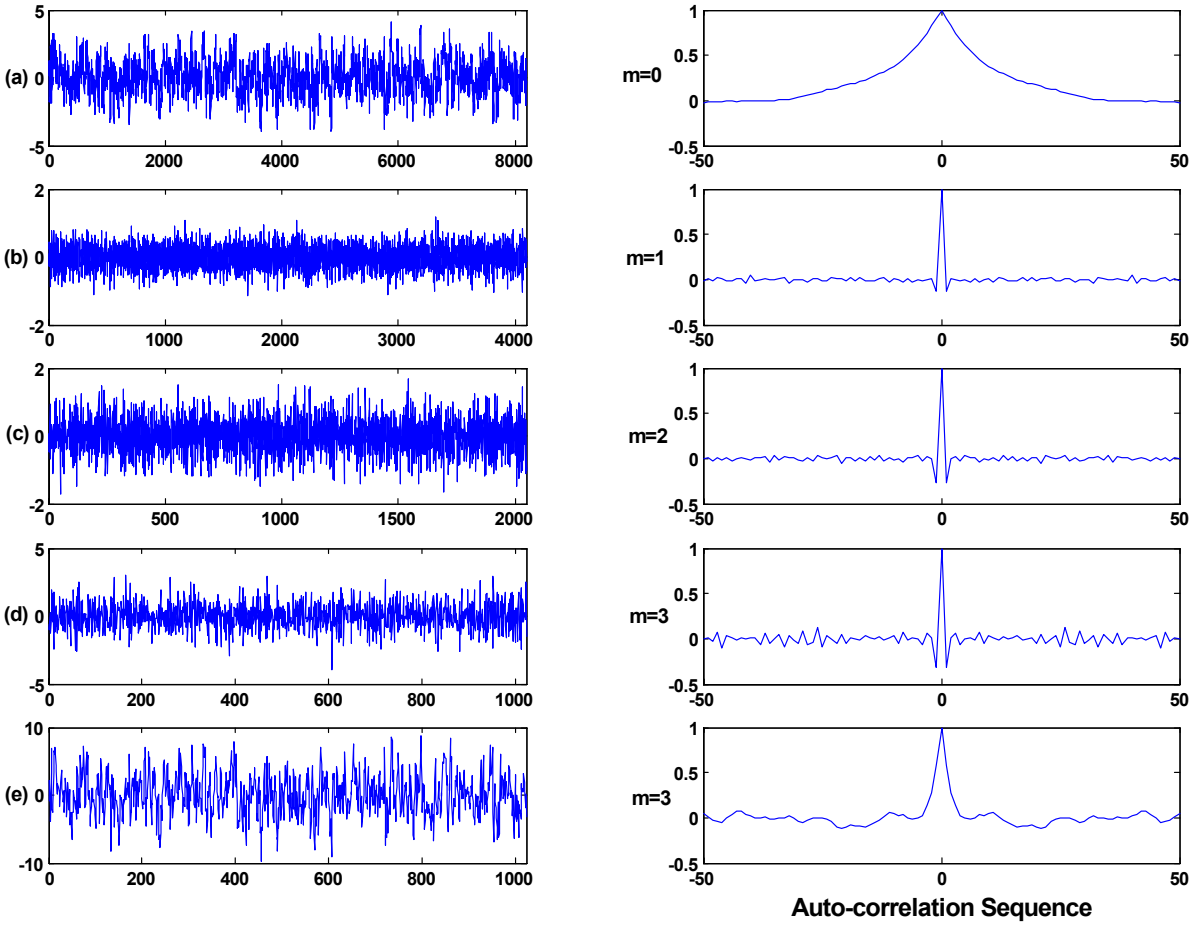

Figure 3.2: Approximate De-correlation due to Dyadic Wavelet Transform

The conventional algorithm for wavelet decomposition is restricted to off-line use since down sampling of the wavelet coefficients causes a time delay in the computation. This limitation may be overcome by decomposing the data in a moving window of dyadic length, with the most recent sample included in the window [113]. The wavelet coefficients obtained by this approach are identical to those obtained by the stationary or non-decimated wavelet transform [106], but the orthonormality of the basis functions and approximate de-correlation of the measurements in each window

are maintained.

3.3.2 Process Monitoring Scheme

3.3.2.1 Methodology

The methodology of MSPCA is shown in Fig. 3.3.

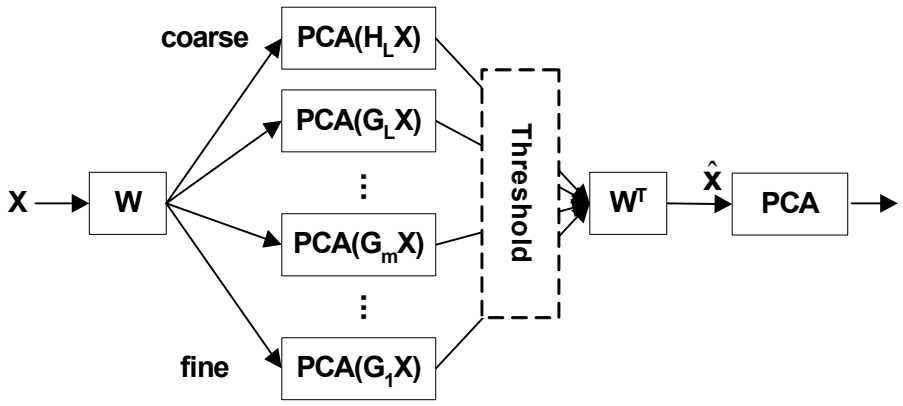

Figure 3.3: MSPCA Methodology

The measurements for each variable are decomposed using the selected family of wavelets. This results in a transformation of the data matrix, X, into a matrix, WX, where W is an orthonormal matrix representing the orthonormal wavelet transformation operator containing the filter coefficients. PCA control charts monitor the resulting coefficients at each scale. The detection limits for T^2 and Q statistics at each scale are determined from the coefficients at that scale of data representing normal operation. Applying separate control charts at each scale permits identification and selection of the scales or frequency bands that contain the significant features representing the current abnormal operation. It allows MSPCA to be more sensitive

to scale-varying signal features such as auto-correlated measurements. The signal is reconstructed from the wavelet and scaling function coefficients larger than the limits at each scale, and PCA is applied again to the reconstructed signal. The control limits for the reconstructed signal can be determined from the limits or variance of the selected scales [3]. This is because the wavelet decomposition decomposes the variance of the data to multiple scales as,

$$X^T X = (H_L X)^T (H_L X) + (G_L X)^T (G_L X) + \ldots + (G_1 X)^T (G_1 X) \tag{3.40}$$

Two important theorems are needed to understand the MSPCA methodology. Their proofs can be found in [3].

Theorem 12 *The principal-component loadings obtained by the PCA of X and WX are identical, whereas the principal component scores of WX are the wavelet transform of the scores of X.*

Theorem 13 *MSPCA reduces to conventional PCA if neither the principal components nor the wavelet coefficients at any scale are eliminated.*

For process monitoring of new data, a statistically significant change is indicated if the T^2 and Q scores based on the wavelet coefficients computed from the most recent measurements violate the detection limits at any scale. Since the wavelet coefficients are sensitive only to changes (see Fig. 3.1(b), (c), (d)), if a variable goes outside the region of normal operation and stays there, the wavelet coefficients will be statistically significant only when the change first occurs. As the change persists, the scaling

coefficients continue violating detection limits for as long as the process remains in an abnormal state (see Fig. 3.1(e)). Similarly, when a process returns from abnormal to normal operation, the change will be detected by the wavelet coefficients, but the scaling coefficients will continue to indicate abnormal operation for several samples due to the coarser scale of representation. Thus, process monitoring based only on PCA of the wavelet and scaling coefficients will not permit quick and continuous detection of a shift, and may create false alarms after a process has returned to normal operation.

Fortunately, the last MSPCA step of selecting the scales that indicate significant events, reconstructing the signal to the time domain, and computing the T^2 or Q scores for the reconstructed signal, improve the speed of detecting abnormal operation and eliminate false alarms after a process returns to normal operation. For statistical process monitoring, those scales are selected at which the wavelet coefficient corresponding to the current measurements violates the detection limits.

Since the reconstructed signal in the time domain is generated from the large wavelet coefficients, MSPCA integrates the task of monitoring with that of extracting the signal features representing abnormal operation, with minimum distortion and time delay. Consequently, there is no need for a separate step for pre-filtering the measured variables. Furthermore, since the covariance matrix for computing the loadings and scores for all scales together depends on the scales at which the current scores or residuals violate the detection limits, the final detection limits for confirming the state of the process also adapt to the nature of the signal features.

For on-line monitoring, the MSPCA algorithm is applied to measurements in a moving window of dyadic length. This approach is based on the online multiscale rectification method in [113], except that instead of rectifying each measured variable, online MSPCA rectifies the T^2 or Q scores computed from the PCA at each scale.

An important practical decision in the MSPCA methodology is that of the depth or number of scales of the wavelet decomposition. Ideally, the depth should be selected to provide maximum separation between the stochastic and deterministic components of a signal. If the depth is too small, then the last scaled signal will have a significant amount of noise that will be retained in the result of MSPCA, but if the depth is too large, the matrix of coefficients at coarser scales will have very few rows due to the dyadic down sampling, and this will affect the accuracy of the PCA at that scale. The reduction in the number of rows is not a limitation for on-line or non-decimated wavelet decomposition. The depth may be determined by cross-validation, but for the illustrative examples in this thesis, a heuristic maximum depth that results in 32 coefficients at the coarsest scale, that is, $L = \log_2 n - 5$ is found to be a good choice.

3.3.2.2 Detailed Procedure

- I. Setup MSPCA Reference Model:

1. Get the reference data when a process is under normal condition. For each variable in the reference data matrix, compute the wavelet decomposition and get the reference wavelet coefficients.

2. For each scale, put the reference wavelet coefficients from all variables together

and apply PCA to get the reference PCA model (including mean, standard deviation and PC loadings) and control limits for T^2 or Q statistics. Repeat this procedure for all scales.

3. Define reconstruction scenarios based on the number of decomposition level. For each reconstruction scenario, assign the selected significant scales with the corresponding reference wavelet coefficients and the insignificant scales with zeros (hard thresholding). Reconstruct the signal from the selected and thresholded coefficients for each variable. Put the reconstructed signal of all variables together and apply PCA to get the reference PCA model (including mean, standard deviation, PC loadings, and control limits) for this reconstruction scenario. Repeat the same procedure to all reconstruction scenarios.

- II. Online Process Monitoring:

1. Determine the size of the moving window of dyadic length, w. Generate a data window with w samples from the real-time data by moving the time window. For each variable in the data window, compute the wavelet coefficients.

2. For each scale, calculate T^2 and Q scores based on the reference PCA model in I.(2).

3. Compare the T^2 and Q scores with the control limits in I.(2), retain wavelet coefficients that violate the control limits and assign those within the control limits to zero.

4. Reconstruct the signal in the moving window variable-by-variable.

5. Since only the most recent sample (the last data in the moving window) is of interest, determine the reconstruction scenario based on the last T^2 and Q scores of each scale and get the detection limits for the last reconstructed signal. Calculate T^2 and Q scores of the last reconstructed signal based on the reference PCA model in I.(3).

6. If T^2 and/or Q scores of the last reconstructed signal exceed the detection limits, fire an alarm.

3.4 Illustrative Examples

3.4.1 Adaptive PCA Test

3.4.1.1 Plant Data Description

The real process data is from the B-4010 pilot plant at the Dow Chemical Company. The pilot plant is used to develop and scale-up new Polypropylene-based products. To support the plant operation and process data analysis, the composition of the vent stream of D-511 is analyzed using a Gas Chromograph (GC). In this example, the following 7 variables were selected in the data set: GC measurement of Propylene, Hydrogen, and Nitrogen, Hydrogen feed, MOD vent production rate, GC residual, and total Propylene feed. The test data includes 4 runs, and its description is listed in Table 3.1.

Table 3.1: Test Data Description

Run	Time Span	Operating Condition
1	5/8/03 8:00 to 5/8/03 16:00	H_2 feed change at 5/8/03 14:40
2	5/10/03 9:00 to 5/11/03 11:00	H_2 feed change at 5/11/03 0:18 & MOD vent production rate change at 5/11/03 0:22
3	5/13/03 15:00 to 5/15/03 21:00	-
4a	5/16/03 21:00 to 5/17/03 5:00	Homopolymer (only propylene/no ethylene)
4b	5/17/03 7:00 to 5/17/03 13:00	Homopolymer (only propylene/no ethylene)

The GC sampling frequency is 2 minutes. Run #4a and #4b are under homopolymer stage, only propylene is involved. Run #1, #2, #3 are under bi-polymer stage, both propylene and ethylene are fed into the reactor. The operating condition is changed in run #1 and #2. Tests are conducted on all 4 runs but only run #1 and #4b test results are listed in the thesis. Two outlier replacement methods, robust outlier replacement and moving median filter, are investigated in both the initialization stage and recursive update stage. A forgetting factor $\beta = 0.98$ is introduced in the recursive calculation of the mean, standard deviation and correlation to ignore old data exponentially. The first 50 samples are used in the initialization stage to build the initial PCA model. The rest of data are used for adaptive PCA test. Test results of conventional PCA and adaptive PCA without forgetting factor are also listed for comparison.

3.4.1.2 Test Result of Run #1

The initial reference data includes 50 samples from 5/8/03 8:00 to 5/8/03 9:38. The test data includes 191 samples from 5/8/03 9:40 to 5/8/03 16:00, where 150

samples are obtained before the operating condition change at 5/8/03 14:40 and 41 samples are taken after the operating condition change. Test results shown in the tables below are listed in the following order, number of samples over control limits, number of false-alarm samples, and number of miss-alarm samples. The best monitoring scheme has the lowest number of false-alarm and miss-alarm points. Two outlier replacement methods are investigated, I represents robust outlier replacement and II represents moving median filter. Test results before the operating condition change are listed in column 'before T' and test results after the operating condition change are listed in column 'after T', where T=5/8/03 14:40. Adaptive PCA monitoring results of run #1 w/w.o. forgetting factor β are shown in Table 3.2. It is easy to see that the best performance is achieved when robust outlier replacement is used and old data is ignored. The performance of Adaptive PCA using the moving median filter is not as good as the robust outlier replacement even in the initialization stage where every sample should be normal. It modifies the reference data more aggressively when the forgetting factor β is employed and tends to adapt to the operating condition change, but results in more miss-alarm points. The purpose of adaptive PCA is accommodating to the normal time-varying process condition, not set point changes. Apparently, robust outlier replacement is a better choice for outlier replacement.

The conventional PCA monitoring result of run #1 is shown in Table 3.3. It is easy to see that conventional PCA fires too many false-alarms when the process is under normal condition. It is overly sensitive to the normal process variation.

Table 3.2: Adaptive PCA Monitoring Result of Run 1

	Initial Data		Test Data			
Outlier Replace	**I**	**II**	**I**		**II**	
			before T	**after T**	**before T**	**after T**
β= 0.98	0/0/0	2/2/0	17/1/0	41/0/0	15/4/5	14/0/27
No β	0/0/0	2/2/0	19/3/0	41/0/0	21/7/2	35/0/6

Table 3.3: Conventional PCA Monitoring Result of Run 1

	Initial Data	Test Data	
		before T	**after T**
Conventional PCA	0/0/0	131/115/0	41/0/0

The T^2 and Q charts of adaptive PCA monitoring with forgetting factor $\beta = 0.98$ are shown in Fig. 3.4 and 3.5. The conventional PCA monitoring result is shown in

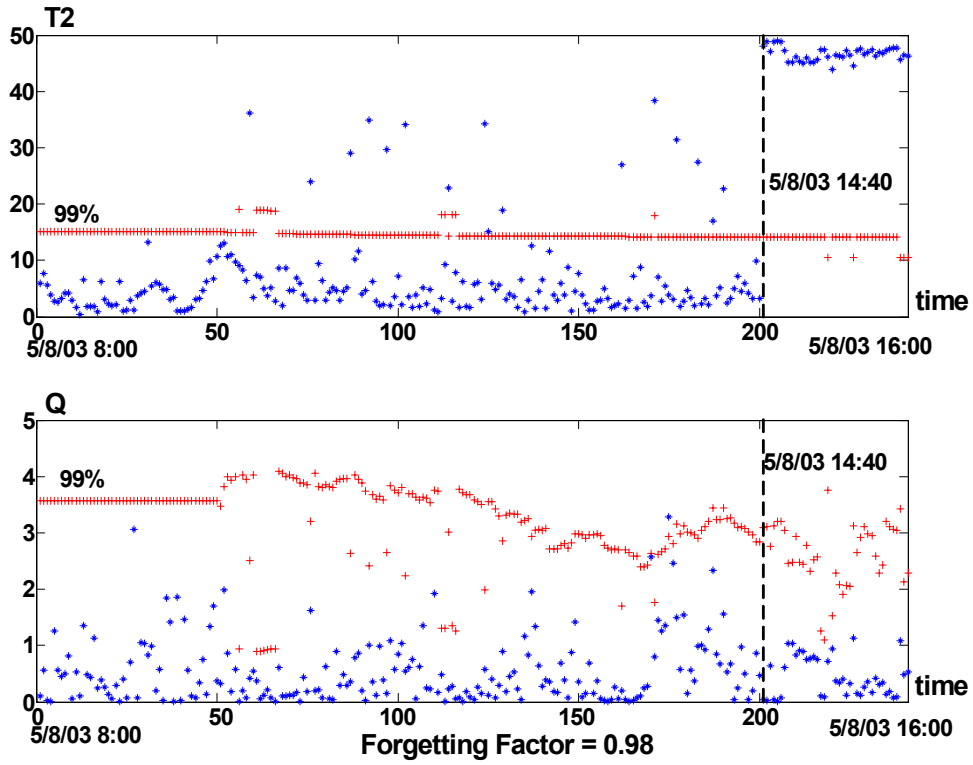

Figure 3.4: Adaptive PCA Monitoring of Run #1 Using Robust Outlier Replacement

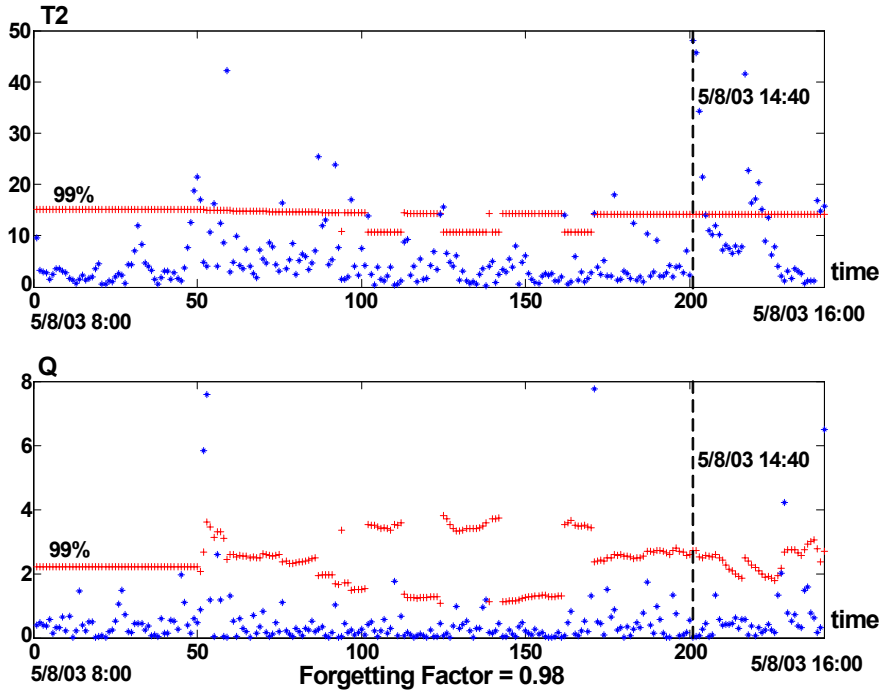

Figure 3.5: Adaptive PCA Monitoring of Run #1 Using Moving Median Filter

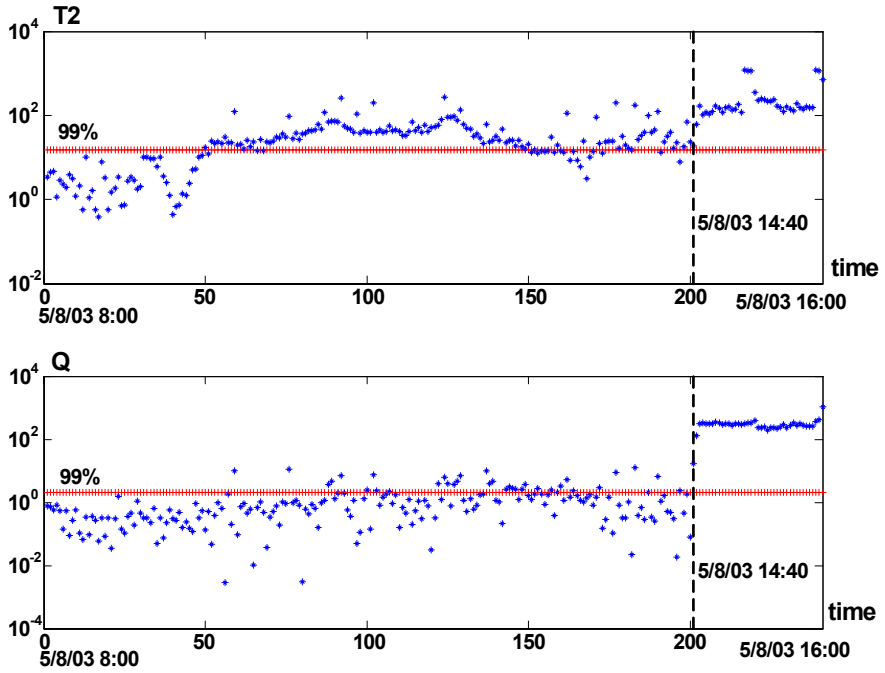

Figure 3.6: Conventional PCA Monitoring of Run #1

Fig. 3.6.

3.4.1.3 Test Result of Run #4b

The initial reference data includes 50 samples from 5/17/03 7:00 - 5/17/03 8:38. The test data includes 131 samples from 5/17/03 8:40 – 5/17/03 13:00. Adaptive PCA monitoring results of run #4b w/w.o. forgetting factor β are shown in Table 3.4. Again, the best performance is achieved when robust outlier replacement is used and old data is ignored.

Table 3.4: Adaptive PCA Monitoring Result of Run 4b

	Initial Data		**Test Data**	
Outlier Replacement	**I**	**II**	**I**	**II**
$\boldsymbol{\beta}$**= 0.98**	0/0/0	3/3/0	4/2/0	6/4/0
No $\boldsymbol{\beta}$	0/0/0	3/3/0	7/5/0	10/8/0

The conventional PCA monitoring result of run #1 is shown in Table 3.5. Again, the conventional PCA fires too many false-alarms when the process is under normal condition.

Table 3.5: Conventional PCA Monitoring Result of Run 4b

	Initial Data	**Test Data**
Conventional PCA	0/0/0	26/24/0

3.4.2 MPCA Simulation Tests

3.4.2.1 Simulation Setup

In this section, MPCA and conventional PCA are applied to a simple 2×2 process to obtain statistically meaningful results and for a fair comparison.

$$\begin{aligned} x(t) &= \begin{bmatrix} 0.118 & -0.191 \\ 0.847 & 0.264 \end{bmatrix} x(t-1) + \begin{bmatrix} 1.0 & 2.0 \\ 3.0 & -4.0 \end{bmatrix} u(t-1) \\ y(t) &= x(t) + v(t) \end{aligned} \tag{3.41}$$

where u is the correlated input,

$$u(t) = \begin{bmatrix} 0.811 & -0.226 \\ 0.477 & 0.415 \end{bmatrix} u(t-1) + \begin{bmatrix} 0.193 & 0.689 \\ -0.320 & -0.749 \end{bmatrix} w(t-1) \tag{3.42}$$

w is an uncorrelated Gaussian signal with zero mean and unit variance. The output y is corrupted by uncorrelated Gaussian errors v with zero mean and variance 0.1, w and v are uncorrelated. The input u and the output y are measured, and those measurements are used for monitoring.

To compare the performance of the monitoring methods, mean shifts of w_1 and changes in the coefficient from u_1 to x_2 are investigated. The settings of those abnormal conditions are summarized in Table 3.6. Note that case 8 simulates a slow drift fault. One data set obtained from the normal operating condition was used to build a PCA model for the conventional PCA and also to determine reference PCs for MPCA. In addition, 200 data sets were used to determine control limits of monitored indices. Different data sets were generated by changing seeds of the white noise

w. Then, 1000 data sets were generated at each case shown in Table 3.6 to test the algorithm.

Table 3.6: Setting of Abnormal Conditions

Case #	**Fault Type**	**Fault Size**
0	Normal Condition	-
1	Mean shift of w_1	$0.0 \rightarrow 1.0$
2	Mean shift of w_1	$0.0 \rightarrow 1.5$
3	Mean shift of w_1	$0.0 \rightarrow 2.0$
4	Mean shift of w_1	$0.0 \rightarrow 3.0$
5	Change of parameter from u_1 to x_2	$3.0 \rightarrow 2.5$
6	Change of parameter from u_1 to x_2	$3.0 \rightarrow 2.0$
7	Change of parameter from u_1 to x_2	$3.0 \rightarrow 1.0$
8	Slow drift of parameter from u_1 to x_2	$3.0 \rightarrow 1.0$

The control limit of each index is determined so that the number of samples outside the control limit is 1% of the entire sample while the process is operated under a normal condition. Therefore, the control limits represent 99% confidence limits without any assumptions of probability distributions of sample statistics. The percentage of the samples outside the control limit is termed 'reliability' and is used for evaluation of each monitoring method.

Based on these control limits, each monitoring method is evaluated by the following steps.

1. Each monitoring method is applied to the data in cases $0 \rightarrow 8$, and each index is calculated.

2. For the data obtained after the occurrence of a fault, the reliability is calculated in each simulation.

3. The average reliability of 1000 simulations is calculated in each case.

Since the control limits are determined so that they represent 99% confidence limits, a monitoring method is regarded as successful in detecting the abnormal condition if the reliability is considerably higher than 1%. On the other hand, when the reliability is less than or close to 1%, the monitoring method is regarded as not functioning well.

3.4.2.2 Monitoring Results and Discussions

In this section, the conventional PCA method is referred to as CPCA. The monitored indices for CPCA are T_i^2 and Q_i. The subscript i denotes the number of adopted PCs. The monitored index for MPCA is A_i or A_{1-m}.

Static Monitoring The vector of variables for static monitoring at step k is the following:

$$\begin{bmatrix} y^T(k) & u^T(k) \end{bmatrix} \tag{3.43}$$

In this thesis, this type of monitoring is called static monitoring to differentiate it from dynamic monitoring, in which past measurements as well as current measurements are used as monitored variables. The static monitoring results of CPCA are summarized in Table 3.7. Two or three PCs are retained in the PCA model. As shown in Table 3.7, the best monitoring performance is achieved when three PCs are used. As expected, the reliability of CPCA increases as the size of the mean shift or

parameter change increases. In almost all cases, however, the reliability measures of CPCA are less than 10%. These results indicate that CPCA does not function well for the changes investigated here. In addition, since the reliability measures in case 0 are very close to 1%, the control limits are successfully determined.

Table 3.7: Reliability (%) of Static CPCA - Applications to the 2 x 2 process

Index	**Control Limit**	**Case**								
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\mathbf{T_2^2}$	9.21	1.1	1.4	2.4	4.4	12.9	0.94	0.86	0.97	0.94
$\mathbf{Q_2}$	4.3693	0.96	1.2	1.9	3.1	8.0	1.2	1.5	2.5	1.5
$\mathbf{T_3^2}$	11.341	0.97	1.4	2.7	5.3	17.1	1.0	1.1	1.8	1.2
$\mathbf{Q_3}$	0.4032	0.95	1.0	1.2	1.5	2.3	1.6	3.0	9.3	3.7

The static monitoring results of MPCA are summarized in Table 3.8. It is amazing that MPCA can outperform CPCA even when only one PC is monitored. For example, the reliability of MPCA in case 7 is 89.5% when A_1 is monitored and $w = 100$, but that of CPCA is less than 10%. Furthermore, the reliability measures of MPCA with A_3 and A_4 are considerably better than the others, especially when $w = 200$, and the detection of abnormal conditions is quite successful. The indices A_3 and A_4 give very nice monitoring results because the data obtained from a normal operating condition can be well explained by two or three PCs. That is, small changes in the third or fourth PC caused by abnormal operations can be easily detected because the changes are quite small when the process is normally operated.

As described above, the variation of process data is well explained by two or three PCs, and abnormal operation can successfully be detected by monitoring the third

Table 3.8: Reliability (%) of Static MPCA - Applications to the 2 x 2 process

Index	**Contr. Limit**	**w**	**Case**								
			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\mathbf{A_1}$	0.2812	100	1.2	0.21	0.19	0.09	0.05	11.7	40.5	89.5	42.0
	0.2220	200	1.2	0.43	1.7	4.9	33.3	10.8	48.1	99.7	51.6
$\mathbf{A_2}$	0.3048	100	1.1	0.26	0.25	0.29	0.78	11.5	45.1	97.1	47.6
	0.2392	200	1.2	0.25	0.72	1.7	7.7	11.0	59.4	99.9	57.8
$\mathbf{A_3}$	0.1511	100	1.5	0.88	0.74	2.8	22.3	1.6	15.6	87.5	25.0
	0.0775	200	1.4	4.8	38.0	87.2	100	20.1	65.8	98.9	63.4
$\mathbf{A_4}$	0.00442	100	1.1	1.2	2.5	4.1	8.2	11.9	77.2	100	62.2
	0.00219	200	1.3	1.7	5.1	13.0	37.8	33.7	99.3	100	87.4
$\mathbf{A_{1-2}}$	0.5288	100	1.5	0.89	0.76	2.8	23.0	1.6	17.9	93.1	28.9
	0.3861	200	1.5	4.8	38.2	87.5	100	20.6	70.4	99.6	66.8

or fourth PC. If so, T_2^2, Q_2, T_3^2 and Q_3 of CPCA should be good indices. However, the reliability measures of those indices shown in Table 3.7 are considerably worse than those of A_3 and A_4 shown in Table 3.8. The reason why the mean shifts and the parameter changes can be detected by MPCA rather than CPCA is that the abnormal operating conditions investigated in this application affect the correlation among variables but the variation of the variables is not increased. The results clearly show the advantages of MPCA over CPCA.

MPCA with A_{1-2} also functions very well. For example, when the subspace spanned by the first two PCs is monitored and the time-window size is 200 steps, MPCA can detect almost all cases except for case 1 successfully. When A_{1-2} is monitored and $w = 100$, MPCA can detect the parameter changes but not the mean shifts. Therefore, the performance of MPCA depends strongly on the design parameters such as the size of the time-window. Note that the index A_{1-3} is not shown in this table

since it yields the same result as A_4. The reason is the total number of variables in this example is 4. Therefore, the index A_{1-3} and A_4 measure the same change of correlation.

Since the reliability measures in case 0 are close to 1%, the control limits are successfully determined so that the probability of false alarms is almost 1% under the normal operating condition. Therefore, the above comparison based on the reliability is sufficiently fair.

Dynamic Monitoring Static monitoring might not function well for auto-correlated data. One approach that has proved useful in dealing with auto-correlated data involves identifying a time-series model such as an autoregressive moving-average (ARMA) model, and then using the model to remove the autocorrelation from the data, and applying PCA charts to the residuals. The use of past measurements as monitored variables is also useful for capturing the autocorrelation and cross-correlation among process variables, because the dynamics can be taken into account [83]. For this order 2 simulation problem, the measurements at one step before are used as monitored variables. The vector of process variables for dynamic monitoring at step k becomes the following:

$$\begin{bmatrix} y^T(k-1) & u^T(k-1) & y^T(k) & u^T(k) \end{bmatrix} \tag{3.44}$$

The dynamic monitoring results of CPCA are summarized in Table 3.9. The best monitoring performance is achieved when six or seven PCs are used. In comparison to the results of static monitoring in Table 3.7, the detection performance, that is,

the reliability, of dynamic monitoring is considerably better. For example, in case 7, the reliability of static CPCA with Q_3 is only 9.3%, but the reliability of dynamic CPCA with Q_6 is 65.6%. Therefore, the superiority of dynamic monitoring over static monitoring is obvious.

Table 3.9: Reliability (%) of Dynamic CPCA - Applications to the 2 x 2 process

Index	**Control Limit**	**Case**								
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\mathbf{T_5^2}$	15.086	0.86	1.3	2.5	5.1	18.4	1.1	1.7	5.1	2.1
$\mathbf{Q_5}$	0.07525	0.78	0.92	1.3	1.8	4.2	2.4	14.6	44.2	17.3
$\mathbf{T_6^2}$	16.812	0.93	1.4	2.6	5.2	18.5	1.5	3.3	12.6	4.5
$\mathbf{Q_6}$	0.01926	0.86	2.4	5.1	10.5	32.5	12.4	38.5	65.6	34.7
$\mathbf{T_7^2}$	18.475	1.0	1.9	4.1	8.8	31.3	7.2	25.0	54.0	25.1
$\mathbf{Q_7}$	0.00929	1.1	3.4	6.6	12.0	30.2	6.9	24.9	53.4	24.9

The dynamic monitoring results of MPCA are summarized in Table 3.10. The best detection performance is achieved when the subspace spanned by the first six PCs is monitored. The performance of dynamic MPCA is better than that of static MPCA. In addition, the performance of dynamic MPCA is considerably better than that of dynamic CPCA when the number of PCs and the time-window size are appropriately selected. These results clearly show the advantages of the proposed method.

3.4.3 MSPCA Simulation Tests

The MSPCA methodology for on-line monitoring is implemented in Matlab based on the wavelet analysis functions available in the software package WaveLab from the website of the statistics department at Stanford university.

Table 3.10: Reliability (%) of Dynamic MPCA - Applications to the 2 x 2 process

Index	**Contr. Limit**	**w**	**Case**								
			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\mathbf{A_{1-4}}$	0.1304	100	1.3	2.8	7.9	15.9	30.4	9.0	83.1	100	62.7
	0.09505	200	0.85	4.6	20.0	47.6	83.9	20.4	99.9	100	85.6
$\mathbf{A_{1-5}}$	0.1225	100	1.1	0.57	1.0	2.1	5.2	0.62	53.5	100	53.6
	0.09141	200	1.1	0.50	1.8	7.7	21.6	0.90	100	100	86.6
$\mathbf{A_{1-6}}$	0.2021	100	0.90	16.4	39.1	49.3	54.1	3.5	31.7	99.9	54.6
	0.1499	200	1.1	32.2	66.2	77.2	83.2	7.6	72.8	100	98.6

3.4.3.1 Mean Shift in Uncorrelated Measurements

This example consists of four variables, with an underlying dimensionality of two. Two variables are represented as uncorrelated Gaussian measurements of zero mean and unit variance, as shown in Eqs. 3.45a and 3.45b. The remaining two variables are formed by adding and subtracting the first two variables, respectively, as shown in Eqs. 3.45c and 3.45d:

$$\widetilde{x}_1(t) = N(0,1) \tag{3.45a}$$

$$\widetilde{x}_2(t) = N(0,1) \tag{3.45b}$$

$$\widetilde{x}_3(t) = \widetilde{x}_1(t) + \widetilde{x}_2(t) \tag{3.45c}$$

$$\widetilde{x}_4(t) = \widetilde{x}_1(t) - \widetilde{x}_2(t) \tag{3.45d}$$

The measured data matrix, X, consists of these four variables contaminated by uncorrelated Gaussian errors of zero mean and standard deviation of 0.2 as, $X(t) = \widetilde{X}(t) + 0.2 \cdot N(0, I)$.

The data representing normal operation consist of 256 equally spaced samples.

The PCA and MSPCA models are developed from these data by selecting the first two principal components. Since the measurements are uncorrelated, the variances at each scale for MSPCA are almost equal. Abnormal operation is indicated by a step change in the mean of all four variables between samples 176 and 225. The performance of MSPCA is illustrated and compared with that of PCA for two different magnitudes of the mean shift.

Case 1 This case considers a mean shift of unity, which is large enough for easy detection by both CPCA and MSPCA. The T^2 and Q charts of CPCA for the abnormal operation are shown in Fig. 3.7. The Q chart easily and unambiguously identifies the mean shift between the samples where it was introduced, but some measurements do go below the 95% detection limit.

The T^2 and Q charts for on-line MSPCA are constructed as illustrated in Fig. 3.8, 3.9, and 3.10. These figures contribute to the final results shown in Fig. 3.11 and 3.12. A sliding window of length 128 is used with wavelet decomposition level $L = 3$. The Q chart at each scale for the wavelet decomposition of data in a window from sample number 49 to 176 is shown in Fig. 3.7(a). Since a step change in the mean contributes to a wide range of frequencies, it is detected by the Q chart at three scales, the wavelet coefficient scale, $m = 1$, $m = 3$, and the scale coefficient scale, $m = 3$, as shown in Fig. 3.8(a). The reconstructed time-domain signal based on the coefficient that violates the 99% limit is shown in Fig. 3.8(b) as a '*'. The 99% detection limit for the most recent measurement is equal to the control limit of the

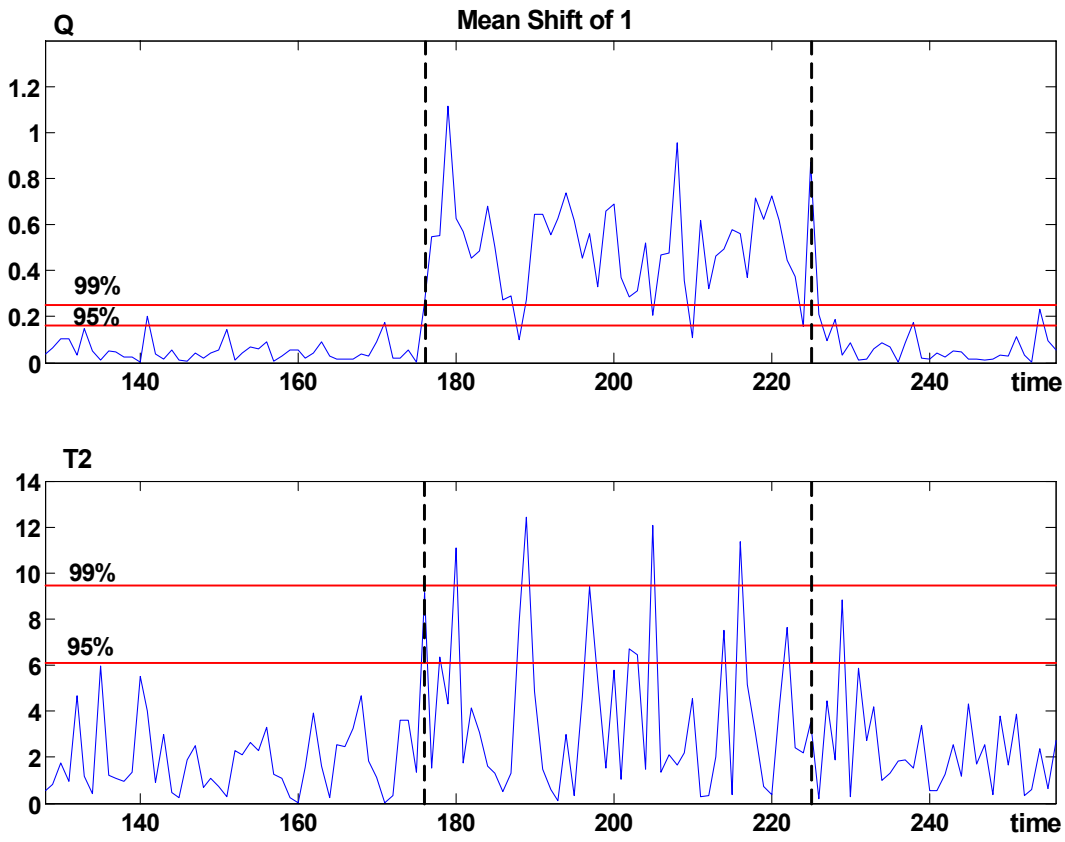

Figure 3.7: CPCA for Detecting Mean Shift of 1 Between Samples [176, 225] in Uncorrelated Measurements

corresponding reconstruction scenario, which can be determined based on the wavelet coefficients of the last sample point in the sliding window. The 99% Q detection limit is shown as a '$\times$' in Fig. 3.8(b). The Q chart for on-line monitoring in Fig. 3.12(a) is constructed by including only the most current value of the reconstructed signal and its detection limit at the sample number 176. Since only the last point is used for on-line monitoring, detection limits are not shown for any other measurements in Fig. 3.8(b).

Detection of the mean shift after it persists for a few samples is illustrated in Fig. 3.9 for data in a window ending at sample 180. The change is no longer detected by

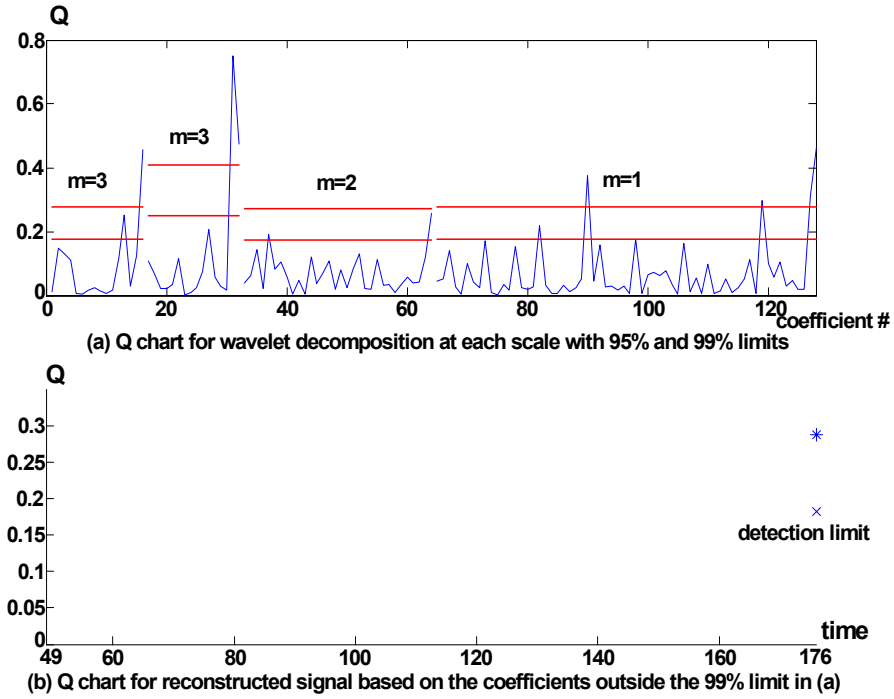

Figure 3.8: Multiscale Detection of the Unit Mean Shift at 176 in Uncorrelated Measurements for Data in Window [49,176]

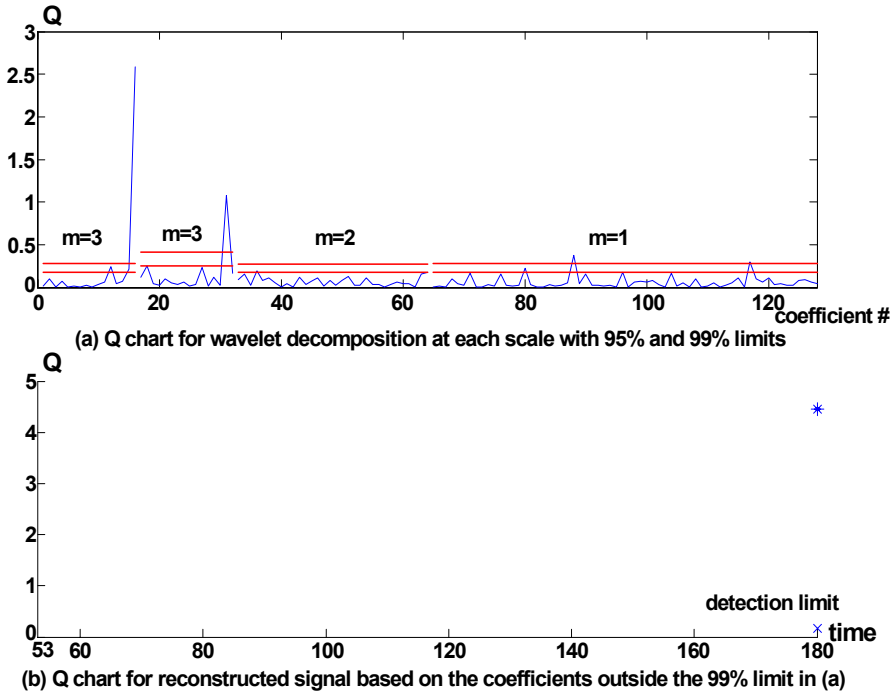

Figure 3.9: Multiscale Detection of the Unit Mean Shift in Uncorrelated Measurements for Data in Window [53,180]

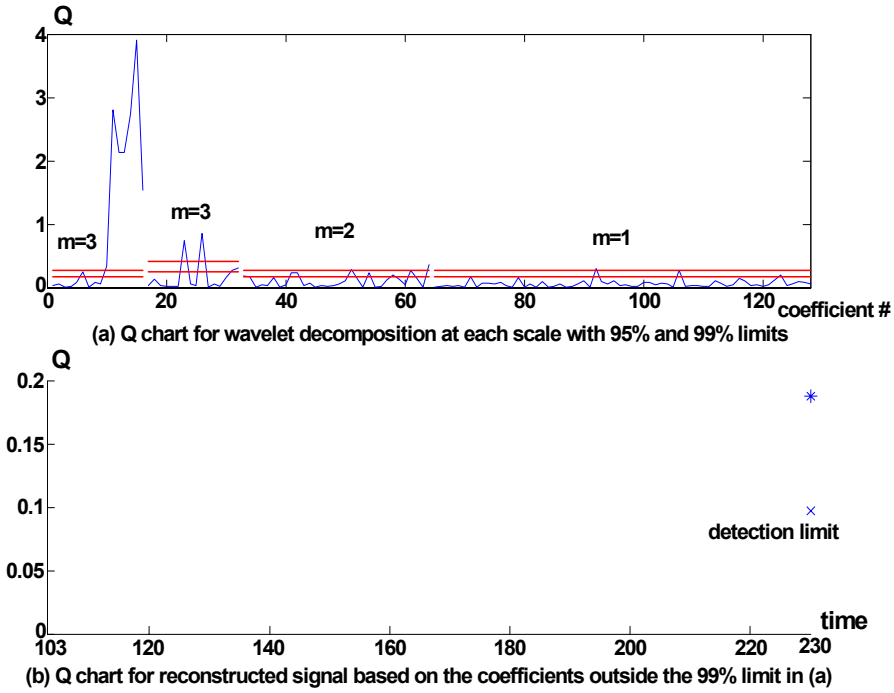

Figure 3.10: Multiscale Detection of the End of the Unit Mean Shift in Uncorrelated Measurements for Data in Window [103,230]

the current coefficients at the finest scale, but is detected by the current coefficients at coarser scales, at $m = 3$, as shown in Fig. 3.9(a). The reconstructed signal based on the coefficients that violate the 99% detection limit in Fig. 3.9(a), and the detection limit for these important scales, are shown in Fig. 3.9(b), and contribute to the signal in Fig. 3.12(a) at sample number 180. The contribution of the change is present at the finer scales, but in older coefficients. While the change continues, it will not be detected by any current wavelet coefficient, but will be detected by the current coefficients in the last scaled signal. If the signal is decomposed to a depth such that the coarsest scaled signal (scale coefficients) contains little contribution from the stochastic process variation, then the current coefficients in the coarsest scaled signal will continue to violate the detection limits after an initial time delay. If the process

returns to normal operation, the change will again be detected first at multiple scales, but the coefficients of the coarsest scaled signal will continue to violate the detection limits for several samples after the process has returned to normal, as shown in Fig. 3.10(a). Thus, any monitoring scheme based only on the wavelet decomposition, without the last MSPCA step of monitoring based on the time-domain reconstructed signal, may not be able to detect a return to normal operation as soon as it occurs. Fig. 3.8(b), 3.9(b), and 3.10(b) demonstrate the ability of MSPCA to extract the feature representing abnormal operation and to adjust the detection limits, depending on the scales at which the abnormal operation is detected.

The T^2 and Q charts for on-line MSPCA are shown in Fig. 3.11 for a confidence limit of 95%, and in Fig. 3.12 for a confidence limit of 99%. Comparison with Fig. 3.7 shows that the Q chart for both confidence limits detects the mean shift at least as accurately as conventional PCA, and all the measurements are well above the confidence limits. The original measured variables and features extracted by MSPCA based on coefficients that violate the 99% limit on the Q chart are shown in Fig. 3.13. These signals are the mean of the reconstructed signals like the ones in Fig. 3.8(b), 3.9(b), and 3.10(b).

Case 2 In this case, the magnitude of the mean shift is 0.3, which is only 30% of the standard deviation of the measurements. This is a pretty challenging problem because of the low signal to noise ratio. The T^2 and Q charts in Fig. 3.14 show that the Q values for the abnormal operation are a little larger in the region of the mean

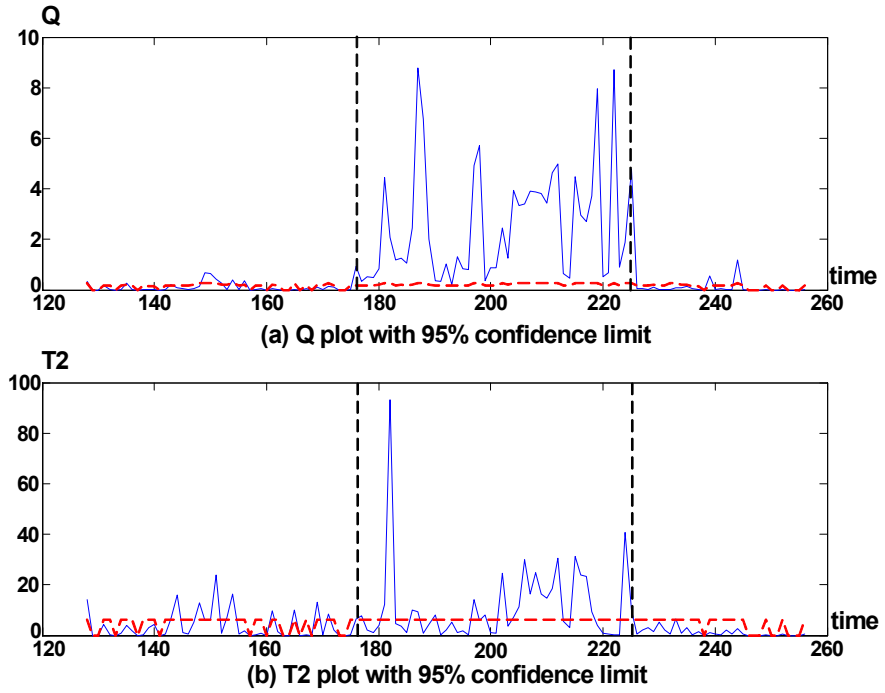

Figure 3.11: MSPCA Monitoring of the Uncorrelated Measurements with Mean Shift of 1

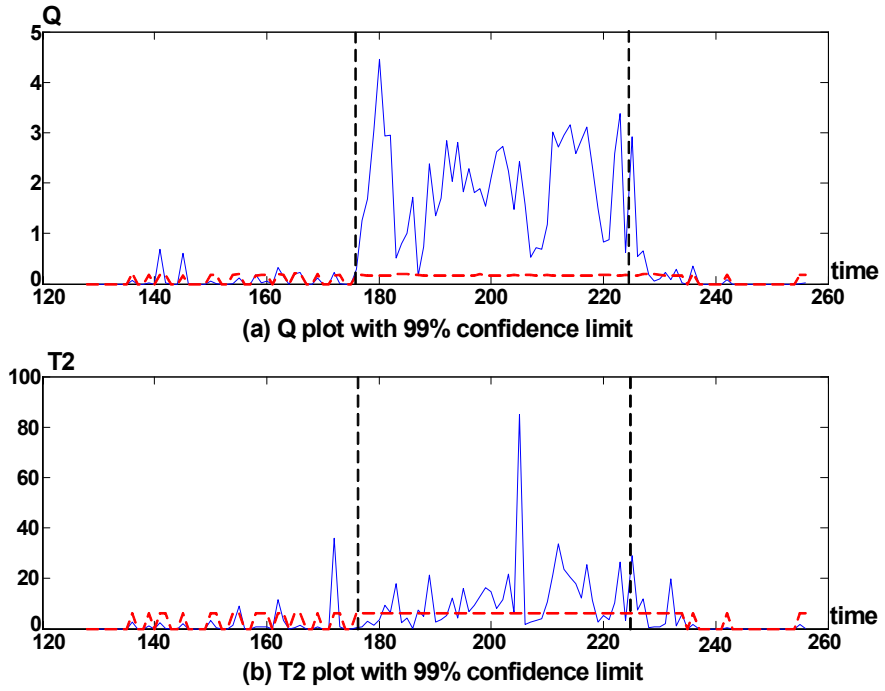

Figure 3.12: MSPCA Monitoring of the Uncorrelated Measurements with Mean Shift of 1

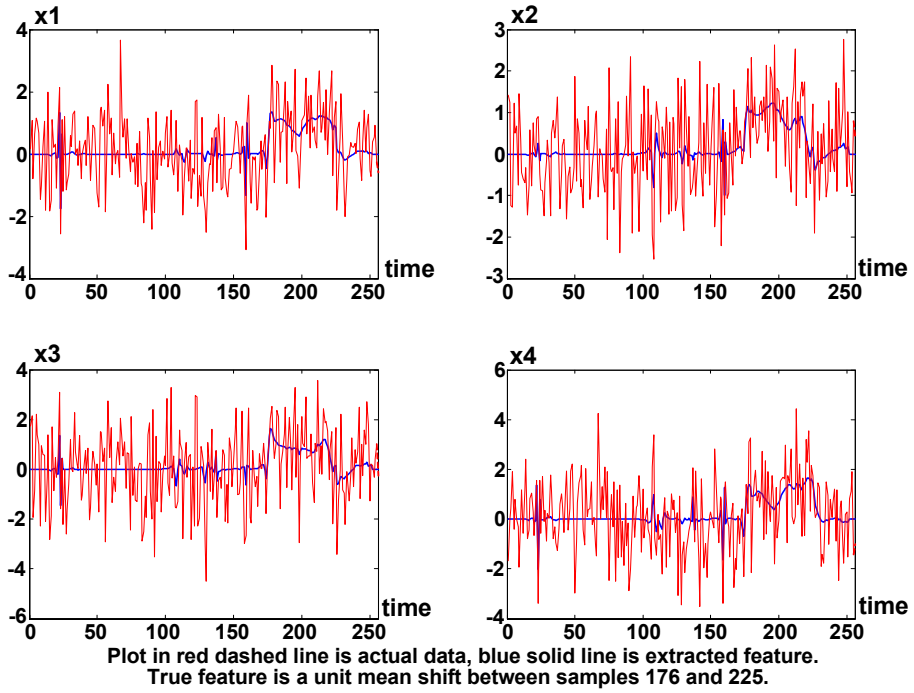

Figure 3.13: Features Relevant to Abnormal Operation Extracted from Each Variable by MSPCA Signal Reconstruction for 99% Confidence

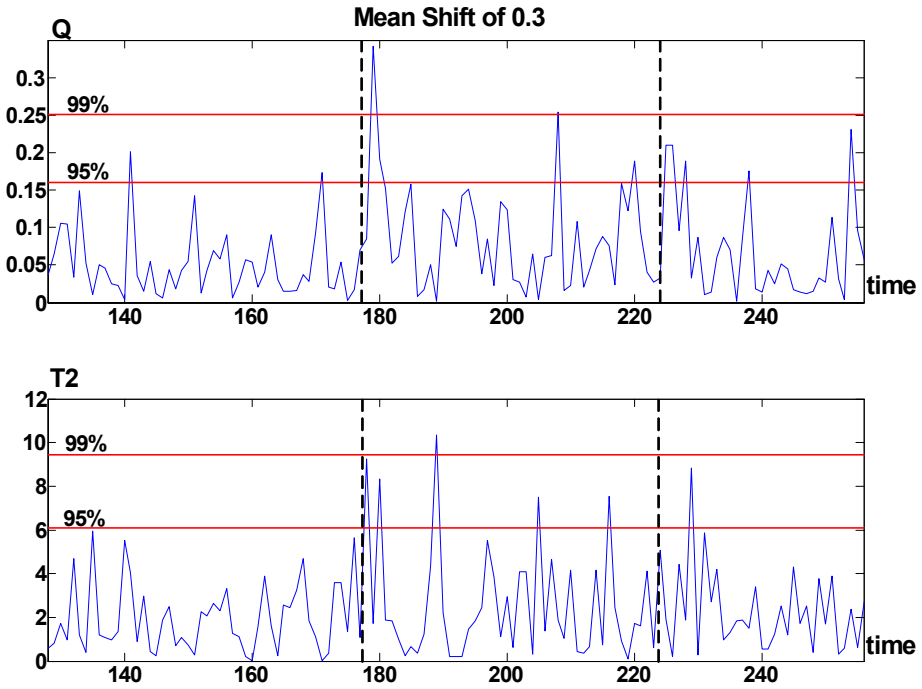

Figure 3.14: CPCA for Detecting Mean Shift of 0.3 Between Samples [176, 225] in Uncorrelated Measurements

shift, but do not consistently violate the confidence limits. Thus, conventional PCA is unable to detect this change. The T^2 and Q charts for on-line MSPCA demonstrate the superiority of monitoring by PCA, as shown in Fig. 3.15 and 3.16. The Q charts for 95% and 99% confidence clearly identify the mean shift over most of the range of abnormal operation, which is significantly better than monitoring by PCA.

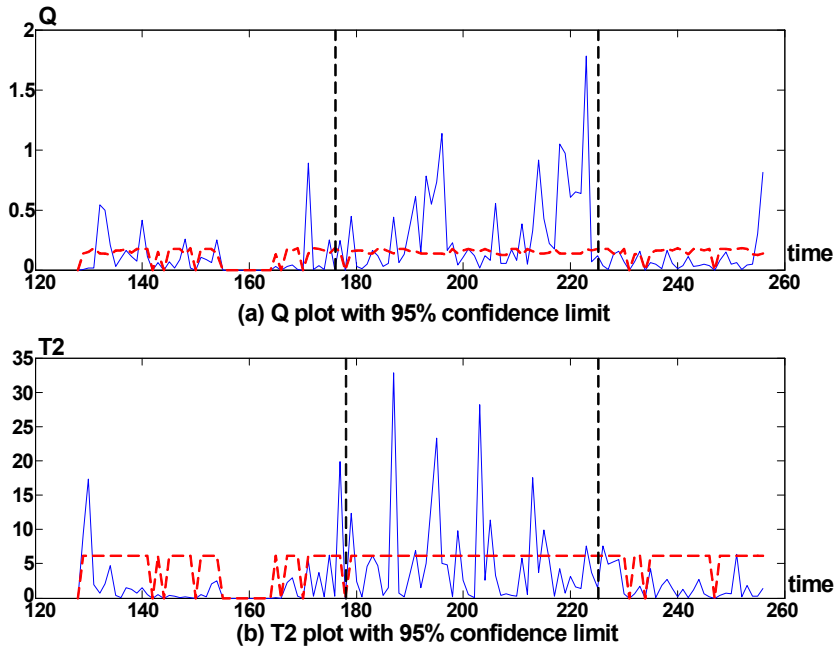

Figure 3.15: MSPCA Monitoring of the Uncorrelated Measurements with Mean Shift of 0.3

3.4.3.2 Monitoring of Auto-correlated Measurements

The data for this example are generated from a model suggested by Ku et al. [83]. The input variables, z and u, are given by

$$\begin{aligned} z(k) &= 0.8 \cdot z(k-1) + u(k-1) \\ u(k) &= 0.7 \cdot u(k-1) + w(k-1) \end{aligned} \tag{3.46}$$

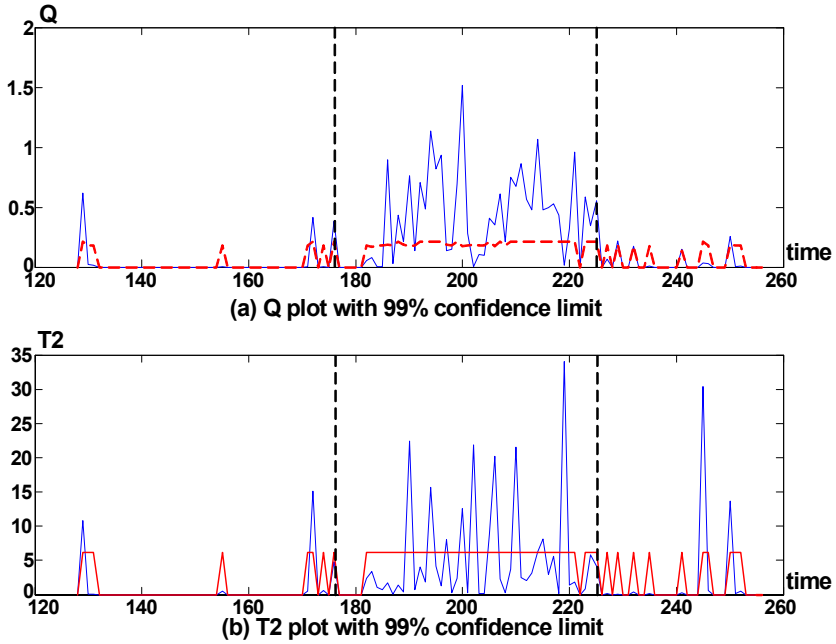

Figure 3.16: MSPCA Monitoring of the Uncorrelated Measurements with Mean Shift of 0.3

where $w(k)$ is white noise with unit variance. The measured data are corrupted by white noise of standard deviation 0.2. The data matrix used for static PCA is given by

$$X(k) = [z(k) \quad u(k)] + 0.2 \cdot N(0, I)$$

The reference data matrix representing normal operation consists of 1,024 measurements. Parallel analysis results in the selection of one principal component. Since the measurements are correlated, the eigenvalues for MSPCA vary with scale according to their power spectrum.

The testing data are generated by introducing a disturbance in the form of a mean shift of magnitude 0.5 in the input variable, u, between samples 201 and 600. The T^2 and Q charts for PCA are shown in Fig. 3.17. The disturbance does affect the

T^2 values, but cannot be detected with reasonable confidence from this chart. The T^2 and Q charts for MSPCA are shown in Fig. 3.18 and 3.19. Clearly, the T^2 chart for MSPCA is significantly better at identifying the disturbance. Fig. 3.18 and 3.19 also show that static MSPCA for auto-correlated measurements does not result in too many false alarms.

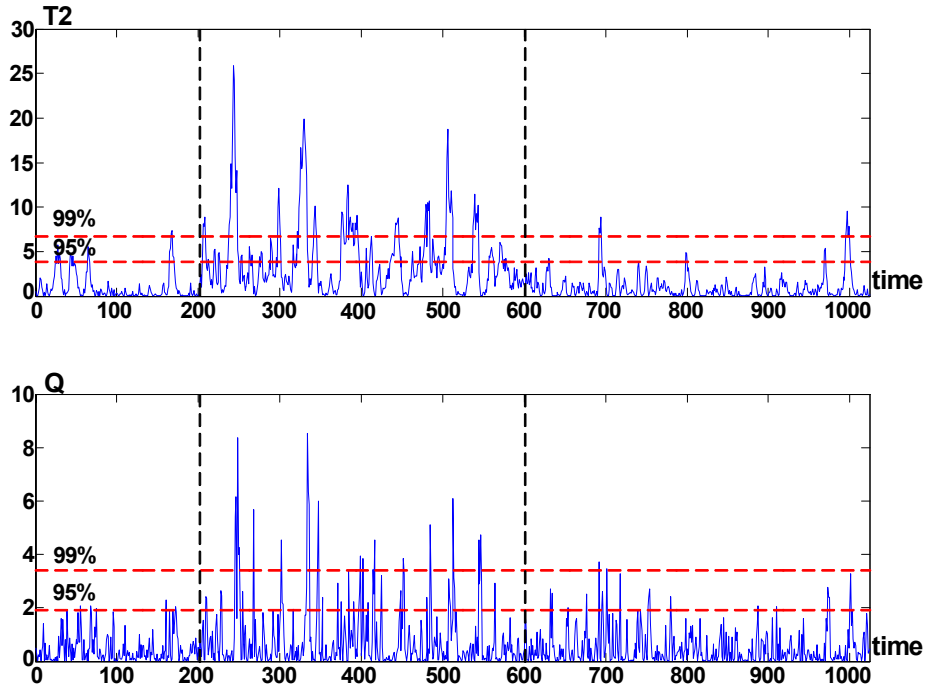

Figure 3.17: Monitoring of Auto-correlated Measurements with Mean Shift of 0.5 by Static PCA

Since the measurements in this example are auto-correlated, the data are also modelled by dynamic PCA by augmenting the data matrix by lagged variables as

$$X(k) = [z(k)\ u(k)\ z(k-1)\ u(k-1)] + 0.2 \cdot N(0, I)$$

Three components are retained by parallel analysis. As shown by the charts in Fig. 3.20, dynamic PCA is also unable to detect the mean shift.

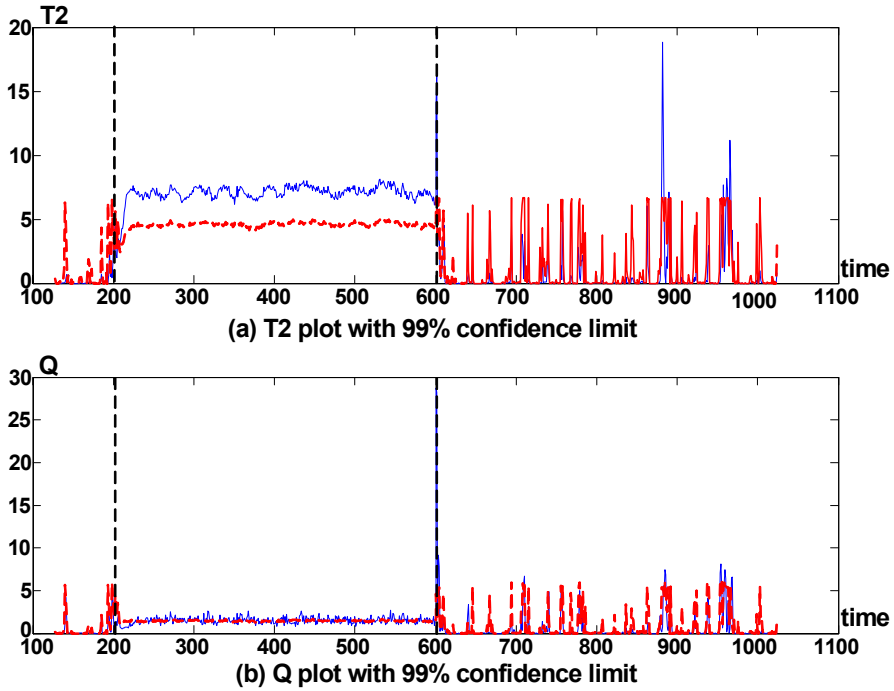

Figure 3.18: Monitoring of Auto-correlated Measurements with Mean Shift of 0.5 by Static MSPCA

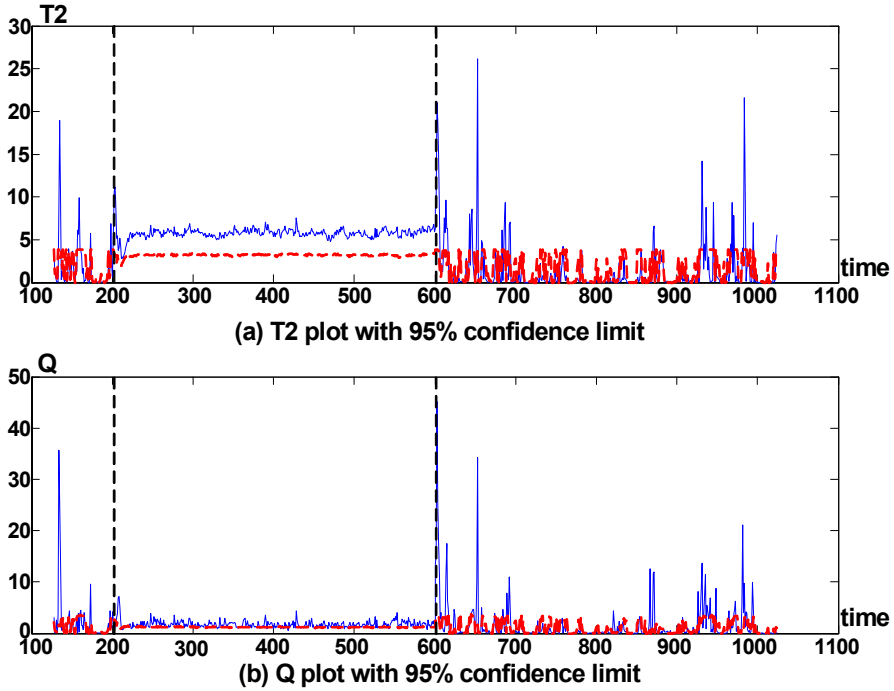

Figure 3.19: Monitoring of Auto-correlated Measurements with Mean Shift of 0.5 by Static MSPCA

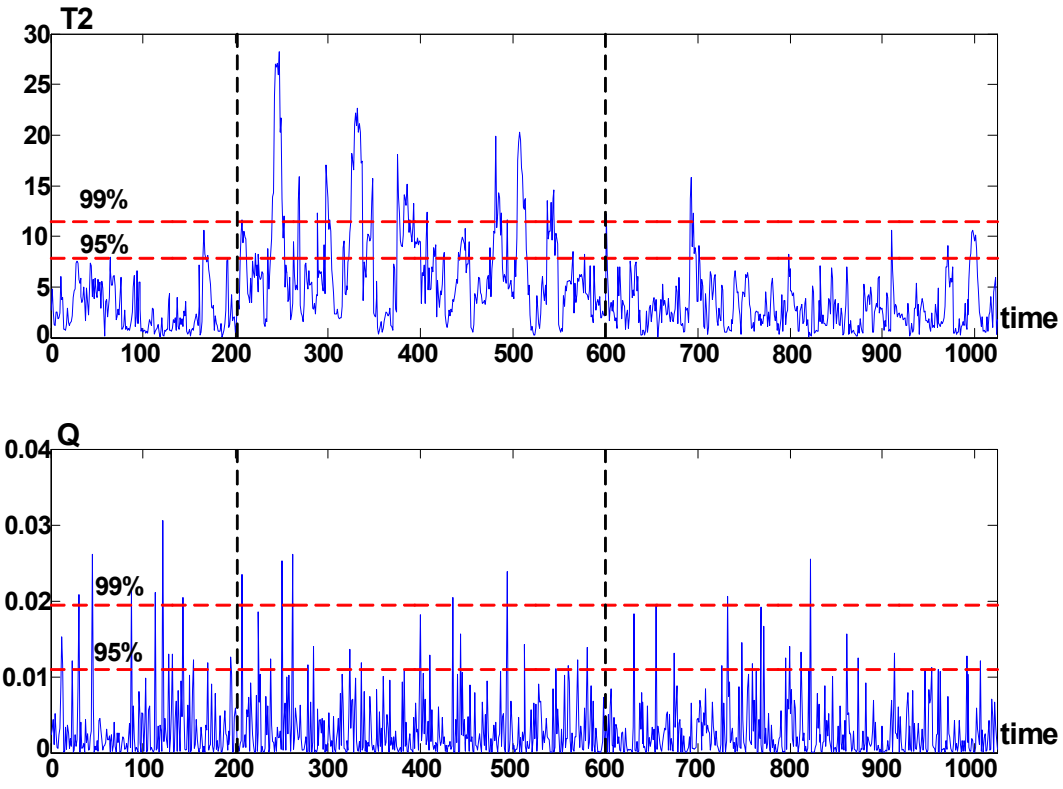

Figure 3.20: Monitoring of Auto-correlated Measurements with Mean Shift of 0.5 by Dynamic PCA

Chapter 4

Conclusion

4.1 Conclusions

This thesis studies two specific approaches, namely the Multiple Model Extended Kalman Filter (MMEKF) and Principal Component Analysis (PCA), under the model based and data driven fault diagnosis categories, respectively.

Chapter 2 describes the MMEKF approach. The MMEKF serves as a residual generator for the fault diagnosis of nonlinear stochastic systems. The MMEKF system consists of a bank of EKFs, where in the absence of modeling errors, only one EKF produces a zero-mean white residual sequence under the normal or a specific fault condition. Therefore, fault detection and isolation (FDI) can be achieved simultaneously. To facilitate the real time monitoring of fault diagnosis, a PCA residual evaluation scheme is proposed. Two multivariate statistical measures, T^2 and Q scores, of the residuals are monitored against their corresponding control limits. The fault decision making is simplified as a limit checking process and this process can easily be automated. After the accomplishment of FDI, a further step is to evaluate the size of the fault. This is in fact a requirement if Fault Diagnosis and Accommoda-

tion (FDA) or Fault Tolerant Control (FTC) strategies are to be applied. The fault magnitude can be estimated by using an efficient finite range search algorithm in the PCA residual evaluation approach.

Simulation study on the CSTR process includes three different kinds of fault conditions, the actuator, sensor and process fault. Satisfactory test results show that the MMEKF fault diagnosis approach has a general application as long as the fault situation can be modelled in the filter properly. Some practical issues for the EKF simulation are discussed. One should note that the applicability of the MMEKF approach depends on observability of the realization, including unmeasurable states and parameters. The parameters of the EKF should be tuned to obtain stable and accurate estimation.

Chapter 3 discusses a very popular topic in the context of data driven fault diagnosis techniques, PCA. Despite its success in industrial process control applications, the conventional PCA approach has its limitations. Three variations of the conventional PCA, namely Adaptive PCA, Moving PCA (MPCA), and Multi-Scale PCA (MSPCA), are presented. Each of these three extensions addresses a certain limitation of conventional PCA. A major limitation of conventional PCA based monitoring is that the PCA model, once built from the training data, is time-invariant, while most real industrial processes are time-varying. An adaptive process monitoring approach is needed to accommodate for the time-varying process condition. When the process is normal but undergoing slow change, Adaptive PCA is preferred.

People tend to overlook the two fundamental statistical assumptions when apply-

ing PCA: measured variables are independent and identically distributed normal random variables; measured variables are uncorrelated in time (not auto-correlated). In practice, measurements from dynamic processes do not satisfy these two assumptions. Research shows that the underlying distribution of data may vary substantially from normality without affecting the results, while auto-correlated measurements would definitely impair PCA performance. The MPCA approach deals with the aspect of the statistical distribution, using a distribution-free multivariate method to describe the probability density function of the measured data. MSPCA combines the ability of PCA to extract the relationship between the variables and de-correlate the cross-correlation between variables with the ability of wavelets to extract features in the measurements and approximately de-correlate the autocorrelation in the measurements. MPCA and MSPCA might be good approaches for detecting slow process drift since they are more sensitive to small change. An industrial application for Adaptive PCA and simulation studies for MPCA and MSPCA show superior performance over conventional PCA.

Note that the residual analyzed in MMEKF approach is assumed to be zero-mean and white for the proper model choice. But uncertainty of the system structure, parameters and the effect of disturbances can cause the residual to deviate from the whiteness assumption substantially, which means that conventional PCA may no longer be suitable for residual evaluation. Fortunately, the extensions of conventional PCA discussed above are expected to solve the problem and improve the robustness of residual evaluation.

4.2 Contributions

The major contributions of this research can be stated as:

- Provide a MMEKF fault diagnosis method to achieve fault detection and isolation of a nonlinear stochastic system.

- Propose a novel PCA residual evaluation approach to monitor the process, and evaluate the fault magnitude by using a finite range search algorithm. The search algorithm is justified by the observation that T^2 and Q statistics of the residual vs EKF fault ratio are unimodal functions.

- Combine the MMEKF framework and the PCA residual evaluation to form a complete fault diagnosis system. This system is applied to the simulation of a CSTR chemical process and satisfactory result is obtained.

- Investigate fault modelling issues in EKF design and some practical issues for the EKF simulation.

- Discuss the consequences of the two fundamental statistical assumptions when applying PCA and investigate the limitations of the conventional PCA. Present three variations of the conventional PCA, namely Adaptive PCA, MPCA, and MSPCA, each addresses a certain limitation of the conventional PCA.

- Point out that Adaptive PCA is better suited in the situation when the process is normal but undergoing slow change which is still considered to be normal,

and MPCA and MSPCA might be good approaches for detecting slow process drift since they are more sensitive to small change.

- Show that the performance of the Adaptive PCA applied to an industrial problem and simulation studies for MPCA and MSPCA is superior over conventional PCA.

- Study some key issues that need to be considered while developing the PCA model, and methodology for Adaptive PCA, MPCA and MSPCA. Derive a sliding window algorithm for MPCA.

4.3 Future Work

Observer-based and PCA approaches themselves are two broad topics in fault diagnosis community and there are many areas of on-going and future work. One of characteristics of PCA is that it is a steady-state (depends on the current operating set point) approach. Whenever there is an operating condition change, the reference PCA model needs to be changed. The solution to this problem is monitoring set-point-free indices instead of original measurement variables. The residual would be an ideal candidate for this index. The PCA is still applicable no matter what kind of model, a mathematical model or a time series model, is used, as long as the residual is not dependent on the set point. Therefore, the fusion of the model based and data driven approaches would be an interesting area to be explored.

The dissertation points out that use of the MPCA and MSPCA approaches instead

of conventional PCA might improve robustness of residual evaluation. Test results need to be shown to support this statement.

The test results of MPCA and MSPCA are based on simulation. Test results based on industrial applications would be more desirable.

There are some parameters in Adaptive PCA, MPCA and MSPCA approaches that need to be tuned by trial and error. An automatic tuning process is more desirable if these approaches are to be applied to industrial process monitoring.

Appendix A

Recursive Updating of Adaptive PCA

A.1 Updating the Standard Deviation

We follow the notations used in Section 3.1.1.1. All the $k+1$ blocks of data, after being mean conferred, are described by

$$X_{k+1}^0 - l_{k+1}\mu_{k+1}^T = \begin{bmatrix} X_k^0 - l_k\Delta\mu_{k+1}^T - l_k\mu_k^T \\ X_{n_{k+1}}^0 - l_{n_{k+1}}\mu_{k+1}^T \end{bmatrix}$$

The standard deviation for the i^{th} variable is

$$\sigma_{k+1.i}^2 = \frac{\left\| \begin{matrix} X_k^0(:,i) - l_k\Delta\mu_{k+1}(i) - l_k\mu_k(i) \\ X_{n_{k+1}}^0(:,i) - l_{n_{k+1}}\mu_{k+1}(i) \end{matrix} \right\|^2}{N_{k+1}-1}$$

and for the i^{th} standard deviation, we are led to

$$(N_{k+1}-1)\sigma_{k+1.i}^2 = \left\|X_k^0(:,i) - l_k\mu_k(i)\right\|^2 + N_k\Delta\mu_{k+1}^2(i) + \left\|X_{n_{k+1}}^0(:,i) - l_{n_{k+1}}\mu_{k+1}(i)\right\|^2$$

where two relationships, $l_k^T\left(X_k^0(:,i) - l_k\mu_k(i)\right) = 0$ and $l_k^T l_k = N_k$ have been applied.

Further, consider the definition that

$$\sigma_{k.i}^2(N_k - 1) = \left\| X_k^0(:, i) - l_k \mu_k(i) \right\|^2$$

we have Eq. 3.11 instantly.

A.2 Updating the Correlation Matrix

Substituting X_{k+1} given by Eq. 3.9 into the definition

$$R_{k+1} = \frac{1}{N_{k+1} - 1} X_{k+1}^T X_{k+1}$$

directly yields

$$\begin{aligned}(N_{k+1} - 1)R_{k+1} &= \Sigma_{k+1}^{-1}\Sigma_k X_k^T \Sigma_k \Sigma_{k+1}^{-1} + l_k^T l_k \Sigma_{k+1}^{-1}\Delta\mu_{k+1}\Delta\mu_{k+1}^T\Sigma_{k+1}^{-1} \\ &\quad -2\Sigma_{k+1}^{-1}\Delta\mu_{k+1}\Delta\mu_{k+1}^T\Sigma_{k+1}^{-1} + X_{n_{k+1}}^T X_{n_{k+1}}\end{aligned}$$

Using the facts that

$$\begin{aligned} l_k^T X_k &= 0 \\ l_k^T l_k &= N_k \\ (N_k - 1)R_k &= X_k^T X_k \end{aligned}$$

in the above equation gives Eq. 3.12 immediately.

Appendix B

Derivation of the Sliding Window Algorithm

A sliding window algorithm is developed to recursively update mean and correlation within the time window.

Suppose at time t_k, the data time window contains samples $x_1 \rightarrow x_w$, $x_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,p}]$ ($i = 1, 2, \ldots, w$) is a p-dimensional vector, where p is the number of variables and w is the window size. At time t_{k+1}, the data time window contains samples $x_2 \rightarrow x_{w+1}$.

The mean at time t_k is

$$\mu_k = \frac{\sum_{i=1}^{w} x_i}{w}$$

And the mean at time t_{k+1} is

$$\mu_{k+1} = \frac{\sum_{i=2}^{w+1} x_i}{w} = \mu_k + \frac{x_{w+1} - x_1}{w}$$

The variance at time t_k is $(w-1)\sigma_{k,j}^2 = \sum_{i=1}^{w} x_{i,j} - \mu_{k,j}$, where $j = 1, 2, \ldots, p$.

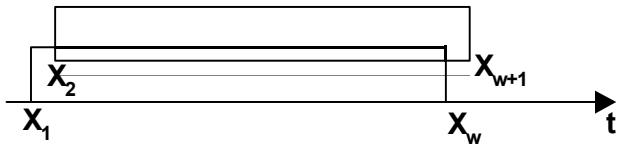

Figure B.1:

And the variance at time t_{k+1} is

$$\begin{aligned}(w-1)\sigma^2_{k+1,j} &= \sum_{i=2}^{w+1}(x_{i,j}-\mu_{k+1,j})^2 = \sum_{i=2}^{w+1}(x_{i,j}-\mu_{k,j}-\frac{x_{w+1,j}-x_{1,j}}{w})^2 \\ &= (w-1)\sigma^2_{k,j}+(x_{w+1,j}-\mu_{k,j})^2-(x_{1,j}-\mu_{k,j})^2-\frac{(x_{w+1,j}-x_{1,j})^2}{w}\end{aligned}$$

Define data matrix at time t_k as $X_k = [y_1\ y_2\ \ldots\ y_p]$, where $y_j = [x_{1,j}\ x_{2,j}\ \ldots\ x_{w,j}]^T$, $j = 1, 2, \ldots, p$, is a column vector. The correlation coefficient at time t_k is

$$\begin{aligned}R^k_{i,j} &= \frac{\sum_{l=1}^{w}(x_{l,i}-\mu_{k,i})(x_{l,j}-\mu_{k,j})}{(w-1)\sigma_{k,i}\cdot\sigma_{k,j}} = \frac{\sum_{l=1}^{w}(x_{l,i}\cdot x_{l,j})-\mu_{k,i}\sum_{l=1}^{w}x_{l,j}-\mu_{k,j}\sum_{l=1}^{w}x_{l,i}+w\cdot\mu_{k,i}\cdot\mu_{k,j}}{(w-1)\sigma_{k,i}\cdot\sigma_{k,j}} \\ &= \frac{\sum_{l=1}^{w}(x_{l,i}\cdot x_{l,j})-w\cdot\mu_{k,i}\cdot\mu_{k,j}}{(w-1)\sigma_{k,i}\cdot\sigma_{k,j}}\end{aligned}$$

The correlation coefficient at time t_{k+1} is

$$\begin{aligned}R^{k+1}_{i,j} &= \frac{\sum_{l=2}^{w+1}(x_{l,i}-\mu_{k+1,i})(x_{l,j}-\mu_{k+1,j})}{(w-1)\sigma_{k+1,i}\cdot\sigma_{k+1,j}} = \frac{\sum_{l=2}^{w+1}(x_{l,i}\cdot x_{l,j})-w\cdot\mu_{k+1,i}\cdot\mu_{k+1,j}}{(w-1)\sigma_{k+1,i}\cdot\sigma_{k+1,j}} \\ &= \frac{(w-1)R^k_{i,j}\cdot\sigma_{k,i}\cdot\sigma_{k,j}+w\cdot\mu_{k,i}\cdot\mu_{k,j}+x_{w+1,i}\cdot x_{w+1,j}-x_{1,i}\cdot x_{1,j}-w\cdot\mu_{k+1,i}\cdot\mu_{k+1,j}}{(w-1)\sigma_{k+1,i}\cdot\sigma_{k+1,j}}\end{aligned}$$

Bibliography

1. H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Contr.* **19**, 716–723 (1974).

2. H. B. Aradhye, B. R. Bakshi, R. A. Strauss, and J. F. Davis. Multiscale SPC using wavelets - theoretical analysis and properties. *AIChE J.* **49**, 939–958 (2003).

3. B. R Bakshi. Multiscale PCA with application to multivariate statistical process monitoring. *AIChE J.* **44**, 1596–1610 (1998).

4. B. R. Bakshi and G. Stephanopoulos. Wave-net: A multiresolution, hierarchical neural network with localized learning. *AICHE J.* **39**(1), 57–81 (1993).

5. M. Basila, G. Jr., Stefanek, and A. Cinar. A model-object based supervisory expert system for fault tolerant chemical reactor control. *Computers and Chem. Eng.* **14**(4-5), 551–560 (1990).

6. M. Basseville. Detecting changes in signals and systems - a survey. *Automatica* **24**(3), 309–326 (1988).

7. M. Basseville. Information criteria for residual generation and fault detection and isolation. *Automatica* **33**(5), 783–803 (1997).

8. M. Basseville and I. V. Nikiforov. "Detection of Abrupt Changes: Theory and Application". Information and System Science, Prentice Hall, New York (1993).

9. G. Bastin and M. R. Gevers. Stable adaptive observers for nonlinear time-varying systems. *IEEE Trans. on Automat. Contr.* pages 650–658 (1988).

10. R. V. Beard. "Failure Accommodation in Linear System Through Self Reorganization". PhD thesis, Massachusetts Institute of Technology (1971).

11. W. Becraft and P. Lee. An integrated neural Network/Expert system approach for fault diagnosis. *Computers and Chem. Eng.* **17**(10), 1001–1014 (1993).

12. G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics* **XLIV**, 141 (1991).

13. S. Bogh. Multiple hypothesis-testing approach to fdi for the industrial actuator benchmark. *Contr. Eng. Practices* **3**(12), 1763–1768 (1995).

14. C. A. Carpenter and S. Grossberg. The art of adaptive pattern recognition by a self-organizing neural network. *Computer 21* **21**(3), 77–88 (1988).

15. R.B. Cattell. The scree test for the number of factors. *Multi- variate Behav. Res.* **1**, 245 (1966).

16. B. H. Chen, X. Z. Wang, S. H. Yang, and C. Mcgreavy. Application of wavelets and neural networks to diagnostic system development. 1. feature extraction. *Computers and Chem. Eng.* **23**(7), 899–906 (1999).

17. J. Chen and R. J. Patton. Optimal filtering and robust fault-diagnosis of stochastic-systems with unknown disturbances. In "IEE Proc.-D: Contr. Theory & Appl.", volume 143, pages 31–36 (1996).

18. J. Chen and R. J. Patton. "Robust Model-Based Fault Diagnosis for Dynamic Systems". Kluwer Academic Publishers, Boston/Dordrecht/London (1999).

19. J. Chen, H. G. Zhang, and H. Y. Zhang. A modified separated-bias estimation approach to the detection and estimation of failures in linear systems. In "Proc. of the 11th IFAC World Congress", Tallin (1990).

20. Q. Chen, R. J. Wynne, P. Goulding, and D. Sandoz. The application of principal component analysis and kernel density estimation to enhance process monitoring. *Contr. Eng. Practice* **8**, 531–543 (2000).

21. J. T. Cheung and G. Stephanopoulos. Representation of process trends part i. a formal representation framework. *Computers and Chem. Eng.* **14**(4-5), 495–510 (1990).

22. E. Y. Chow and A. S. Willsky. Issues in the development of a general algorithm for reliable failure detection. In "Proc. of the 19th Conf. on Decision & Contr.", Albuquerque, NM (1980).

23. E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. on Automat. Contr.* (7), 603–614 (1984).

24. C. K. Chui and G. Chen. "Kalman Filtering with Real-Time Applications". Springer Series in Information Sciences. Springer-Verlag, 2 edition (1991).

25. R. N. Clark. The dedicated observer approach to instrument failure detection. In "Proc. of The 18th IEEE Conf. on Decision & Contr.", pages 237–241, Fort Lauderdale, FL, USA (1979).

26. R. Da. Failure detection of dynamical systems with the state χ^2 test. *J. of Guidance, Contr. & Dynamics* **17**(2), 271–277 (1994).

27. R. Da and C. Lin. Sensor failure detection with a bank of kalman filters. In "Proc. of 1995 Amer. Contr. Conf.", pages 1122–1126, Seattle, WA, USA (1995).

28. K. C. Daly, E. Gai, and J. V. Harrison. Generalized likelihood test for fdi in redundancy sensor configurations. *J. of Guidance, Contr. & Dynamics* **2**(1), 9–17 (1979).

29. I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications of Pure and Applied Mathematics* **XII**, 909 (1988).

30. J. de Kleer and S. Brown. A qualitative physics based on confluences. *Artificial Intelligence* **24**(1-3), 7–83 (1984).

31. M. Desai and A. Ray. A fault detection and isolation methodology - theory and application. In "Proc. 1984 Amer. Contr. Conf.", pages 262–270, San Diego, USA (1984).

32. R. W. Dijkerman and R. R. Majumdar. Wavelet representations of stochastic processes and multiresolution stochastic models. *IEEE Trans. Signal Processing* **42**, 1640 (1994).

33. X. Ding and P. M. Frank. Fault detection via factorization approach. *Syst. Contr. Lett.* **14**(5), 431–436 (1990).

34. D. Dong and T. J. McAvoy. Batch tracking via nonlinear principal component analysis. *AICHE J.* **42**(8), 2199–2208 (1996).

35. D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: Asymptopia? *J. of the Royal Statist. Society- Series B* **57**, 41 (1995).

36. F. Doymaz, A. Bakhtazad, J. A. Romagnoli, and A. Palazoglu. Wavelet-based robust filtering of process data. *Computers and Chem. Eng.* **25**, 1549–1559 (2001).

37. F. Doymaz, J. Chen, J. A. Romagnoli, and A. Palazoglu. A robust strategy for real-time process monitoring. *J. of Process Contr.* **11**, 343–359 (2001).

38. R. Dunia and S. J. Qin. A subspace approach to multidimensional fault identification and reconstruction. *AIChE J.* **44**, 1813–1831 (1998).

39. R. Dunia, S. J. Qin, T.F. Edgar, and T.J. McAvoy. Identification of faulty sensors using principal component analysis. *AICHE J.* **42**(10), 2797–2812 (1996).

40. P. Eide and P. S. Maybeck. An mmae failure detection system for the f-16. *IEEE Trans. on Aerospace and Electronic Systems* **32**(3), 1125–1136 (1996).

41. B. Falkenhainer and K. Forbus. Compositional modeling: Finding the right model for the job. *Artificial Intelligence* **51**, 95–143 (1991).

42. J. Y. Fan, M. Nikolaou, and R. E. White. An approach to fault diagnosis of chemical processes via neural networks. *AICHE J.* **39**(1), 82–88 (1993).

43. A. E. Farell and S. D. Roat. Framework for enhancing fault diagnosis capabilities of artificial neural networks. *Computers and Chem. Eng.* **18**(7), 613–635 (1994).

44. P. M. Frank. Advanced fault detection and isolation schemes using nonlinear and robust observers. In "10th IFAC World Congress", Munich, Germany (1987).

45. P. M. Frank. Enhancement of robustness in observer-based fault detection. In "Proc. IFAC/IMACS Sympo. SAFEPROCESS'91", volume 1, pages 275–287, Baden-Baden (1991).

46. P. M. Frank and X. Ding. Frequency domain approach to optimally robust residual generation anf evaluation for model-based fault diagnosis. *Automatica* **30**(4), 789–804 (1994).

47. P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *J. of Process Contr.* **7**(6), 403–424 (1997).

48. B. Friedland and S. Grabousky. Estimating sudden changes of biases in linear dynamic systems. *IEEE Trans. on Automat. Contr.* **27**(1), 237–240 (1982).

49. J. B. Fussell. Fault tree analysis - state of the art. *IEEE Trans. on Reliability* **23**(1), 51–53 (1974).

50. V.B. Gallagher, R.M. Wise, S.W. Butler, D.D. White, and G.G. Barna. Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process; improving robustness through model updating. In "Proc. Of ADCHEM 97", pages 78–83, Ban., Canada (June 1997).

51. A. Genovesi, J. Harmand, and J. P. Steyer. A fuzzy logic based diagnosis system for the on-line supervision of an anaerobic digestor pilot-plant. *Biochemical Eng. J.* **3**(3), 171–183 (1999).

52. J. Gertler. Fault detection and isolation using parity relations. *Contr. Eng. Practice* **5**(5), 653–661 (1997).

53. J. Gertler, Q. Luo, K. Anderson, and X. W. Fang. Diagnosis of plant failures using orthogonal parity equations. In "Proc. of the 11th IFAC World Congress", Tallin (1990).

54. J. Gertler and R. Monajemy. Generating directional residuals with dynamic parity relations. *Automatica* **31**(4), 627–635 (1995).

55. J. Gertler and D. Singer. A new structural framework for parity equation-based failure detection and isolation. *Automatica* **26**(2), 381–388 (1990).

56. J. J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Contr. Syst. Mag.* **8**(6), 3–11 (1988).

57. S. D. Grantham and L. H. Ungar. A first principles approach to automated troubleshooting of chemical plants. *Computers and Chem. Eng.* **14**(7), 783–798 (1990).

58. S. D. Grantham and L. H. Ungar. Comparative analysis of qualitative models when the model changes. *AICHE J.* **37**(6), 931–943 (1991).

59. F. Hamelin and D. Sauter. Robust residual generation for fdi in uncertain dynamic systems. In "Proc. 34th IEEE Conf. on Decision & Contr.", New Orleans, USA (1995).

60. F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. Stahel. "Robust Statistics: The Approach Based on Influence Functions". Wiley, New York (1986).

61. C. Han, R. Shih, and L. Lee. Quantifying signed directed graphs with the fuzzy set for fault diagnosis resolution improvement. *Indust. and Eng. Chemistry Research* **33**(8), 1943–1954 (1994).

62. E.J. Henley. Application of expert systems to fault diagnosis. In "AIChE Annual Meeting", San Francisco, CA (1984).

63. K.A. Hoo, K.J. Tvarlapati, M.J. Piovoso, and R. Hajare. A method of robust multivariate outlier replacement. *Computers and Chem. Eng.* **26**, 17–39 (2002).

64. J. C. Hoskins, K. M. Kaliyur, and D. M. Himmelblau. Fault diagnosis in complex chemical plants using artificial neural networks. *AICHE J.* **37**(1), 137–141 (1991).

65. M. Iri, K. Aoki, E. OShima, and H. Matsuyama. An algorithm for diagnosis of system failures in the chemical process. *Computers and Chem. Eng.* **3**(1-4), 489–493 (1979).

66. Y. Iwasaki and H. A. Simon. Causality in device behavior. *Artificial Intelligence* **29**(1), 3–32 (1986).

67. J. E. Jackson. "A User's Guide to Principal Components". Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc. (1991).

68. C. A. Jacobson and C. N. Nett. An integrated approach to controls and diagnostics using the four parameter control. *IEEE Contr. Syst. Mag.* **11**(6), 22–29 (1991).

69. M. Janusz and V. Venkatasubramanian. Automatic generation of qualitative description of ProcessTrends for fault detection and diagnosis. *Eng. Applications of Artificial Intelligence* **4**(5), 329–339 (1991).

70. R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME J. Basic Eng.* **83**, 95–108 (1961).

71. M. Kano, S. Hasebe, I. Hashimoto, and Hiromu Ohno. A new multivariate statistical process monitoring method using principal component analysis. *Computers and Chem. Eng.* **25**, 1103–1113 (2001).

72. M. H. Kaspar and W. H. Ray. Chemometric methods for process monitoring and high performance controller design. *AICHE J.* **38**(10), 1593–1608 (1992).

73. S. N. Kavuri and V. Venkatasubramanian. Representing bounded fault classes using neural networks with ellipsoidal functions. *Computers and Chem. Eng.* **17**(2), 139–163 (1993).

74. S. N. Kavuri and V. Venkatasubramanian. Using fuzzy clustering with ellipsoidal units in neural networks for robust fault classification. *Computers and Chem. Eng.* **17**(8), 765–784 (1993).

75. S. N. Kavuri and V. Venkatasubramanian. Neural network decomposition strategies for large scale fault diagnosis. *Int. J. of Contr.* **59**(3), 767–792 (1994).

76. H. Kay and B. Kuipers. Numerical behavior envelopes for qualitative models. In "Proceedings of AAAI-93", pages 606–613, Menlo Park, CA (1993).

77. J. Y. Keller, L. Summerer, M. Boutayeb, and M. Darouach. Generalized likelihood ratio approach for fault detection in linear dynamic stochastic systems with unknown inputs. *Int. J. of Syst. Sci.* **27**(12), 1231–1241 (1996).

78. B. E. Kelly and F. P. Lees. The propagation of faults in process plants: 2. fault tree synthesis. *Reliability Eng.* **16**(1), 39–62 (1986).

79. M. Kokawa, M. Satoshi, and S. Shigai. Fault location using digraph and inverse direction search with application. *Automatica* **19**(6), 729–735 (1983).

80. M. A. Kramer and B. L. Palowitch. A rule based approach to fault diagnosis using the signed directed graph. *AIChE J.* **33**(7), 1067–1078 (1987).

81. J. V. Kresta, J. F. MacGregor, and T. E. Marlin. Multivariate statistical monitoring of process operating performance. *Can. J. of Chem. Eng.* **69**(1), 35 (1991).

82. W. J. Krzanowski. Between-groups comparison of principal components. *J. Amer. Statist. Assoc.* **74**, 703–707 (1979).

83. W. Ku, R. H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamical principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **30**, 179–196 (1995).

84. B. Kuipers. Qualitative simulation. *Artificial Intelligence* **29**(3), 289–338 (1986).

85. S. A. Lapp and G. A. Powers. Computer-aided synthesis of fault trees. *IEEE Trans. on Reliability* **26**(1), 2–13 (1977).

86. J. A. Leonard and M. A. Kramer. Diagnosing dynamic faults using modular neural nets. *IEEE Expert* **8**(2), 44–53 (1993).

87. D. Leung and J. Romagnoli. Dynamic probabilistic model-based expert system for fault diagnosis. *Computers and Chem. Eng.* **24**(11), 2473–2492 (2000).

88. R. Li and X. Wang. Qualitative/Quantitative simulation of process temporal behavior using clustered fuzzy digraphs. *AIChE J.* **47**(4), 906–919 (2001).

89. W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin. Recursive PCA for adaptive process monitoring. *J. of Process Contr.* **10**, 471–486 (2000).

90. K. A. Loparo, M. L. Adams, W. Lin, M. F. Abdel-Magied, and N. Afshari. Fault detection and diagnosis of rotating machinery. *IEEE Trans. on Indust. Electronics* **47**(5), 1005–1014 (2000).

91. K. A. Loparo, M. R. Buchner, and K. S. Vasudeva. Leak detection in an experimental heat exchanger process: A multiple model approach. *IEEE Trans. on Automat. Contr.* **36**(2), 167–177 (1991).

92. K. A. Loparo, Z. R. Roth, and S. J. Eckert. Nonlinear filtering for systems with random structure. *IEEE Trans. on Automat. Contr.* **31**(1), 1064–1067 (1986).

93. R. Luo, M. Misra, S. J. Qin, R. Barton, and D. M. Himmelblau. Sensor fault detection via multiscale analysis and dynamic PCA. *Ind. Eng. Chem. Res.* **37**, 1489–1495 (1999).

94. W. L. Luyben. "Process Modeling, Simulation, and Control for Chemical Engineers." New York: McGraw-Hill (1990).

95. J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Contr. Eng. Practice* **3**(3), 403–414 (1995).

96. D. T. Magill. Optimal adaptive estimation of samples stochastic processes. *IEEE Trans. on Automat. Contr.* (4), 434–439 (1965).

97. Y. Maki and K. A. Loparo. A neural-network approach to fault detection and diagnosis in industrial processes. *IEEE Trans. on Contr. Syst. Technology* **5**(6), 529–541 (Nov 1997).

98. E. R. Malinowski. "Factor Analysis in Chemistry". Wiley-Interscience: New York (1991).

99. S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **11**, 674 (1989).

100. H Martens and T. Nas. "Multivariate Calibration", page 159. Wiley, New York (1989).

101. M. A. Massoumnia and W. E. Vander Velde. Generating parity relations for detecting and identifying control system component failures. *J. of Guidance, Contr. & Dynamics* **11**(1), 60–65 (1988).

102. M. Mavrovouniotis and G. Stephanopoulos. Reasoning with order of magnitudes and approximate relations. In "Proceedings of AAAI-87" (July 1987).

103. P. S. Maybeck and P. D. Hanlon. Performance enhancement of a multiple model adaptive estimator. In "Proc. 32nd IEEE Conf. on Decision & Contr.", pages 462–468, San Antonio, Texas, USA (1993).

104. R. K. Mehra and J. Peschan. An innovation approach to fault detection and diagnosis systems. *Automatica* **7**, 637–640 (1971).

105. R. Milne. Strategies for diagnosis. *IEEE Trans. on Syst., Man and Cyber.* **17**(3), 333–339 (1987).

106. G. P. Nason and B. W. Silverman. "Wavelets and Statistics", chapter The Stationary Wavelet Transform and Some Statistical Applications. Springer-Verlag, New York (1995).

107. A. Negiz and A. Cinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AICHE J.* **45**, 1041 (1997).

108. P. R. C. Nelson, P. A. Taylor, and J. F. MacGregor. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems* **35**, 45–65 (1996).

109. K. Niida. Expert system experiments in processing engineering. In "Inst. Of Chem. Eng. Symposium Series", pages 529–583 (1985).

110. I. Nikifirov, V. Varavva, and V. Kireichikov. Application of statistical fault-detection algorithms to navigation systems monitoring. *Automatica* **29**(5), 1275–1290 (1993).

111. R. Nikoukhah. Innovations generation in the presense of unknown inputs - applications to robust failure-detection. *Automatica* **30**(12), 1851–1867 (1994).

112. P. Nomikos and J. MacGregor. Monitoring batch processes using multiway principal component analysis. *AICHE J.* **40**(8), 1361–1375 (1994).

113. M. N. Nounou and B. R. Bakshi. Online multiscale filtering of random and gross errors without process models. *AIChE J.* **45**, 1041 (1999).

114. H. Ocak. "Fault Detection, Diagnosis and Prognosis of Rolling Element Bearings: Frequency Domain Methods and Hidden Markov Modeling". PhD thesis, Case Western Reserve University (2003).

115. J. O'Reilly. "Observers for Linear Systems". Academic Press (1983).

116. D. W. Osten. Selection of optimal regression models via cross- validation. *J. of Chemometrics* **2**, 39–48 (1988).

117. O. O. Oyeleye and M. A. Kramer. Qualitative simulation of chemical process systems: Steady state analysis. *AICHE J.* **34**(9), 1441–1454 (1988).

118. R. J. Patton. Fault-tolerant control: the 1997 situation (survey). In "Proc. IFAC Sympo. on Fault Detection, Supervision and Safety for Technical Processes: SAFEPROCESS'97", pages 1029–1052, Univ. of Hull, UK (1997).

119. R. J. Patton. Robustness in model-based fault diagnosis: the 1995 situation. *A. Rev. Contr.* **21**, 103–123 (1997).

120. R. J. Patton and J. Chen. A review of parity space approaches to fault diagnosis. In "Proc. IFAC/IMACS Sympo. SAFEPROCESS'91", volume 1, pages 239–255, Baden-Baden (1991).

121. R. J. Patton and J. Chen. Observer-based fault detection and isolation: Robustness and applications. *Contr. Eng. Practice* **5**(5), 671–682 (1997).

122. R. J. Patton and S. M. Kangethe. "Fault Diagnosis in Dynamic Systems, Theory and Application", chapter 4, pages 99–154. Prentice Hall (1989).

123. M.J. Piovoso, K.A. Kosanovich, and J. P. Yuk. Process data chemometrica. *IEEE Trans. on Instrumentation and Measurement* **41**(2), 262–268 (1992).

124. S. Qin and W. Li. Detection, identification, and reconstruction of faulty sensors with maximized sensitivity. *AICHE J.* **45**(9), 1963–1976 (1999).

125. S. J. Qin. Recursive PLS algorithms for adaptive data modeling. *Computers and Chem. Eng.* **22**, 503–514 (1998).

126. S. J. Qin and T. J. McAvoy. Nonlinear PLS modeling using neural networks. *Computers and Chem. Eng.* **16**(4), 379–391 (1992).

127. T. E. Quantrille and Y. A. Liu. "Artificial Intelligence in Chemical Engineering". Academic Press, San Diego, LA (1991).

128. A. Raich and A. Cinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AICHE J.* **42**(4), 995–1009 (1996).

129. A. Raich and A. Cinar. Diagnosis of process disturbances by statistical distance and angle measures. *Computers and Chem. Eng.* **21**(6), 661–673 (1997).

130. O. Raiman. Order of magnitude reasoning. In "Proceedings of AAAI-86", pages 100–104 (August 1986).

131. T. S. Ramesh, J. F. Davis, and G. M. Schwenzer. Knowledge-based diagnostic systems for continuous process operations based upon the task framework. *Computers and Chem. Eng.* **16**(2), 109–127 (1992).

132. T. S. Ramesh, S. K. Shum, and J. F. Davis. A structured framework for efficient problem-solving in diagnostic expert systems. *Computers and Chem. Eng.* **12**(9-10), 891–902 (1988).

133. S. Rannar, J.F. MacGregor, and S. Wold. Adaptive batch monitoring using hierarchical PCA. In "AIChE Annual Meeting", Los Angeles, CA (November 1997).

134. A. Ray and R. Luck. An introduction to sensor signal validation in redundant measurement systems. *IEEE Contr. Syst. Mag.* **11**(2), 44–49 (1991).

135. G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. "Engineering Optimization: Methods and Applications". Wiley, New York (1983).

136. R. Rengaswamy, T. Hagglund, and V. Venkatasubramanian. A qualitative shape analysis formalism for monitoring control loop performance. *Eng. Applications of Artificial Intelligence* **14**(1), 23–33 (2001).

137. R. Rengaswamy and V. Venkatasubramanian. A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Eng. Applications of Artificial Intelligence* **8**(1), 35–51 (1995).

138. R. Rengaswamy and V. Venkatasubramanian. A fast training neural network and its updation for incipient fault detection and diagnosis. *Computers and Chem. Eng.* **24**(2-7), 431–437 (2000).

139. A. Rigopoulos, Y. Arkun, F. Kayihan, and E. Hanezyc. Identification of paper machine full profile disturbances models using adaptive principal component analysis. In "Chemical Process Control: V", pages 275–279, Tahoe (January 1996).

140. J. Rissanen. Modeling by shortest data description. *Automatica* **14**, 465–471 (1978).

141. C. RojasGuzman and M. A. Kramer. Comparison of belief networks and rule-based systems for fault diagnosis of chemical processes. *Eng. Applications of Artificial Intelligence* **3**(6), 191–202 (1993).

142. E. Sacks. Qualitative analysis of piecewise linear approximation. *J. of Artificial Intelligence in Eng.* **3**(3), 151–155 (1988).

143. M. S. N. Saïd, M. E. H. Benbouzid, and A. Benchaib. Detection of broken bars in induction motors using an extended kalman filter for rotor resistance sensorless estimation. *IEEE Trans. on Energy Conversion* **15**(1), 66–70 (2000).

144. B. M. Sadler and A. Swami. Analysis of multiscale products for step detection and estimation. *IEEE Trans. on Information Theory* **45**, 1043 (1999).

145. N. J. Scenna. Some aspects of fault diagnosis in batch processes. *Reliability Eng. and Syst. Safety* **70**(1), 95–110 (2000).

146. R. Seliger and P. M. Frank. Fault diagnosis by disturbance decoupled nonlinear observers. In "Proc. 30th IEEE Conf. on Decision & Contr.", volume 3, pages 2248–2253, Brighton, England (1991).

147. R. Seliger and P. M. Frank. Robust residual evaluation by threshold selection and a performance index for nonlinear observer-based fault diagnosis. In "Int. Conf. on Fault Diagnosis (Tooldiag'93)", Toolouse, France (1993).

148. J. Shiozaki, H. Matsuyama, E. OShima, and M. Iri. An improved algorithm for diagnosis of system failures in the chemical process. *Computers and Chem. Eng.* **9**(3), 285–293 (1985).

149. J. Shiozaki, H. Matsuyama, K. Tano, and E. OShima. Fault diagnosis of chemical processes by the use of signed, directed graphs. extension to five-range patterns of abnormality. *Int. Chem. Eng.* **25**(4), 651–659 (1985).

150. S. Soylemez and W. D. Seider. A new technique for precedence-ordering chemical process equation sets. *AICHE J.* **19**(5), 934–942 (1973).

151. J. L. Speyer and J. E. White. Shiryayev sequential probability ratio test for redundancy management. *J. of Guidance, Contr. & Dynamics* **7**(5), 588–595 (1984).

152. E. Tarifa and N. Scenna. Fault diagnosis, directed graphs, and fuzzy logic. *Computers and Chem. Eng.* **21**, S649–654 (1997).

153. N. D. Tracy and J. C. Young. Multivariate control charts for individual observations. *J. of Quality Technology* **24**(2), 88–95 (April 1992).

154. A. B. Trunov and M. M. Polycarpou. Robust fault diagnosis of state and sensor faults in nonlinear multivariable systems. In "Proc. of 1999 Amer. Contr. Conf.", pages 608–612, San Diego, CA, USA (1999).

155. C. S. Tsai and C. T. Chang. Dynamic process diagnosis via integrated neural networks. *Computers and Chem. Eng.* **19**, s747–s752 (1995).

156. S. G. Tzafestas and K. Watanabe. Modern approaches to system/sensor fault detection and diagnosis. *Journal A* **31**(4), 42–57 (1990).

157. N. H. Ulerich and G. A. Powers. Online hazard aversion and fault diagnosis in chemical processes: The digraph /fault tree method. *IEEE Trans. on Reliability* **37**(2), 171–177 (1988).

158. T. Umeda, T. Kuriyama, E. Oshima, and H. Matsuyama. A graphical approach to cause and effect analysis of chemical processing systems. *Chem. Eng. Science* **35**(12), 2379–2388 (1980).

159. L. H. Ungar, B. A. Powell, and S. N. Kamens. Adaptive networks for fault diagnosis and process control. *Computers and Chem. Eng.* **14**(4-5), 561–572 (1990).

160. R. Vaidhyanathan and V. Venkatasubramanian. Digraph-based models for automated HAZOP analysis. *Reliability Eng. and Syst. Safety* **50**(1), 33–49 (1995).

161. R. Vaidyanathan and V. Venkatasubramanian. Representing and diagnosing dynamic process data using neural networks. *Eng. Applications of Artificial Intelligence* **5**(1), 11–21 (1992).

162. S. Valle, W. Li, and S. J. Qin. Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Ind. Eng. Chem. Res.* **38**, 4389–4401 (1999).

163. D. van Schrick. Investigation of reliability for instrument fault detection state-estimator schemes. *European J. of Diagnosis and Safety in Automation* **1**(1), 63–78 (1991).

164. H. Vedam and V. Venkatasubramanian. Signed digraph based multiple fault diagnosis. *Computers and Chem. Eng.* **21**, S655–S660 (1997).

165. H. Vedam and V. Venkatasubramanian. A wavelet theory-based adaptive trend analysis system for process monitoring and diagnosis. In "American Contr. Conf.", pages 309–313 (1997).

166. H. Vedam, V. Venkatasubramanian, and M. Bhalodia. A b-spline based method for data compression, process monitoring and diagnosis. *Computers and Chem. Eng.* **22**, S827–S830 (1998).

167. V. Venkatasubramanian. CATDEX, knowledge-based systems in process engineering: Case studies in heuristic classification. Technical Report, The CACHE Corporation, Austin, TX (1989).

168. V. Venkatasubramanian and K. Chan. A neural network methodology for process fault diagnosis. *AICHE J.* **35**(12), 1993–2002 (1989).

169. V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis part i: Quantitative model-based methods. *Computers and Chem. Eng.* **27**, 293–311 (2003).

170. V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis part II: Qualitative models and search strategies. *Computers and Chem. Eng.* **27**, 313–326 (2003).

171. V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis part III: Process history based methods. *Computers and Chem. Eng.* **27**, 327–346 (2003).

172. V. Venkatasubramanian, R. Vaidyanathan, and Y. Yamamoto. Process fault detection and diagnosis using neural networks i: Steady state processes. *Computers and Chem. Eng.* **14**(7), 699–712 (1990).

173. M. Vidyasagar. "Control System Synthesis: A Factorization Approach", volume 8 of "North Holland System and Control Series". MIT Press, Cambridge, MA, USA (1985).

174. H. Wang, Song Z., and P. Li. Statistical process monitoring with measured data corrupted by noise and gross error. In "Proc. Of the 3rd World Congress on Intelligent Contr. and Automation", pages 680–684 (2000).

175. X. Z. Wang, B. H. Chen, S. H. Yang, and C. Mcgreavy. Application of wavelets and neural networks to diagnostic system development. 2. an integrated framework and its application. *Computers and Chem. Eng.* **23**(7), 945–954 (1999).

176. K. Watanabe, S. Hirota, L. Iloa, and D. M. Himmelblau. Diagnosis of multiple simultaneous fault via hierarchical artificial neural networks. *AICHE J.* **40**(5), 839–848 (1994).

177. K. Watanabe, I. Matsura, M. Abe, M. Kubota, and D. M. Himmelblau. Incipient fault diagnosis of chemical processes via artificial neural networks. *AICHE J.* **35**(11), 1803–1812 (1989).

178. M. Wax and T. Kailath. Detection of signals by information theoretic criterial. *IEEE Trans. Acoust. Speech, Signal Processing* **33**, 387–392 (1985).

179. J. R. Whiteley and J. F. Davis. A similarity-based approach to interpretation of sensor data using adaptive resonance theory. *Computers and Chem. Eng.* **18**(7), 637–661 (1994).

180. N. A. Wilcox and D. M. Himmelblau. Possible cause and effect graphs (PCEG) model for fault diagnosis i: Methodology. *Computers and Chem. Eng.* **18**(2), 103–116 (1994).

181. N. A. Wilcox and D. M. Himmelblau. Possible cause and effect graphs (PCEG) model for fault diagnosis II: Applications. *Computers and Chem. Eng.* **18**(2), 117–127 (1994).

182. A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica* **12**(6), 601–611 (1976).

183. A. S. Willsky, J. J. Deyst, and B. S. Crawford. Adaptive filtering and self-test methods for failure detection and compensation. In "Proc. of 1974 Joint Amer. Contr. Conf.", pages 637–645, Austin, USA (1974).

184. A. S. Willsky and H. L. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Trans. on Automat. Contr.* (2), 108–121 (1976).

185. B. M. Wise and N. B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *J. of Process Contr.* **6**(6), 329–348 (1996).

186. B. M. Wise, N. L. Ricker, D. F. Veltkamp, and B. R. Kowalski. A theoretical basis for the use of principal component models for monitoring multivariate process. *Process Contr. and Quality* **1**, 41–51 (1990).

187. M. Wo, W. Gui, D. Shen, and Y. Wang. Export fault diagnosis using role models with certainty factors for the leaching process. In "Proc. Of the Third World Congress on Intelligent Contr. and Automation", volume 1, pages 238–241, Hefei, China (28 June-2 July 2000).

188. S. Wold. Cross validatory estimation of the number of components in factor and principal component analysis. *Technometrics* **20**, 397–406 (1978).

189. S. Wold. Exponentially weighted moving principal component analysis and projection to latent structures. *Chemometrics and Intelligent Laboratory Systems* **23**, 149–161 (1994).

190. N. E. Wu. Failure sensitizing reconfigurable control design. In "Proc. 31st IEEE Conf. on Decision & Contr.", pages 44–49, Tucson, AZ, USA (1992).

191. G. Xu and T. Kailath. Fast estimation of principal eigenspace using lanczos algorithm. *SIAM J. Matrix Anal. Appl.* **15**, 974–994 (1994).

192. H. Zhang, A. K. Tangirala, and S. L. Shah. Dynamic process monitoring using multiscale PCA. In "Proc. Of 1999 Canadian Conf. On Electrical and Comput. Eng.", pages 1579–1584 (1999).

193. J. Zhang and P. Roberts. Process fault diagnosis with diagnostic rules based on structural decomposition. *J. of Process Contr.* **1**(5), 259–269 (1991).

194. J. Zhao, B. Chen, and J. Shen. A hybrid ANN-ES system for dynamic fault diagnosis of hydrocracking process. *Computers and Chem. Eng.* **21**, S929–S933 (1997).

195. A. Zolghadri. An algorithm for real-time failure-detection in kalman filters. *IEEE Trans. on Automat. Contr.* **41**(10), 1537–1539 (1996).