Steven X. Ding

Advanced Methods for Fault Diagnosis and Fault-tolerant Control

Springer

Advanced Methods for Fault Diagnosis and Fault-tolerant Control

Steven X. Ding

Advanced Methods for Fault Diagnosis and Fault-tolerant Control

Springer

Steven X. Ding
Universität Duisburg-Essen
Duisburg, Germany

ISBN 978-3-662-62003-8 ISBN 978-3-662-62004-5 (eBook)
https://doi.org/10.1007/978-3-662-62004-5

Responsible Editor: Michael Kottusch
This Springer imprint is published by the registered company Springer-Verlag GmbH, DE part of Springer Nature.
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

To My Parents and Eve Limin

Preface

This book is the third one in my book series plan. While the first two are dedicated to model-based and data-driven fault diagnosis respectively, this one addresses topics in both model-based and data-driven thematic fields, and increasingly focuses on fault-tolerant control issues and application of machine learning methods.

The enthusiasm for machine learning and big data technologies has considerable influences on the development of fault diagnosis techniques in recent years. It seems that research efforts in the thematic domain of data-driven fault diagnosis gradually become a competition under the Olympic motto, *faster* transferring machine learning methods to fault diagnosis applications, preferably adopting *higher* actual (most popular) machine learning methods, and *stronger* publishing. The main intention of this book is to study *basic* fault diagnosis and fault-tolerant control problems, which build a framework for *long-term* research efforts in the fault diagnosis and fault-tolerant control domain. In this framework, possibly *unified* solutions and methods can be developed for general classes of systems.

This book is composed of six parts. Besides Part I, which serves as a common basis for the subsequent studies, Parts II–VI are dedicated to five different thematic areas. In Part II, optimal fault detection and estimation in time-varying systems, detection and isolation of multiplicative faults in linear time-invariant systems with uncertainties are addressed. Part III is dedicated to the investigation on existence conditions of observer-based fault detection systems for a general type of nonlinear systems, as well as on parameterisation and optimisation issues of nonlinear observer-based fault detection systems. Part IV deals with statistical and data-driven fault diagnosis, but is dedicated to different topics, including a critical review of multivariate analysis based fault detection methods, optimal fault detection and estimation in large-scale distributed and interconnected systems, Kullback-Leibler divergence based fault detection schemes, and alternative fault detection and clustering methods using symmetric positive definite data matrices and based on Riemannian manifold theory. In Part V, the well-established randomised algorithm theory is applied to the study on assessment and design of fault diagnosis systems. Finally, fault-tolerant control schemes with a strong focus on performance degradation monitoring and recovering are studied in Part VI. These parts are self-contained and so structured that they can also be used for self-study on the concerned topics.

It should be mentioned that the final work on this book has been done during the Corona crisis. I was so deeply sorry to hear of Dr. Jie Chen's death due to the coronavirus. Jie was a good friend, a pioneer and a brilliant researcher of our community. He will be sadly missed.

This book would not be possible without valuable support from many people. I would like to thank Prof. Dr.-Ing. L. Li from the University of Science and Technology Beijing for the long-term collaboration and for the extensive editorial corrections of the book. I am very grateful to my Ph.D. students and co-workers for the valuable discussions and proofreading of the book chapters. They are Ms. Caroline Charlotte Zhu, Ms. Ting Xue, Ms. Han Yu, Ms. Yuhong Na, Mr. Yannian Liu and Mr. Jiarui Zhang.

Finally, I would like to express my gratitude to Mrs. Hestermann-Beyerle and Mrs. Lisa Burato from Springer-Verlag. Mrs. Hestermann-Beyerle has initiated this book project and Ms. Lisa Burato has perfectly managed the final submission issues.

Duisburg Steven X. Ding
May 2020

Contents

Part I Introduction, Basic Concepts and Preliminaries

Part II Fault Detection, Isolation and Estimation in Linear Dynamic Systems

Part IV Statistical and Data-driven Fault Diagnosis Methods

Notation

$\forall$	for all
$\in$	belong to
$\subset$	subset
$\cup$	union
$\cap$	intersection
$\equiv$	identically equal
$\approx$	approximately equal
$A := B$, $B =: A$	A is defined as B
$\Rightarrow$	implies
$\Leftrightarrow$	equivalent to
>> (<<)	much greater (less) than
max (min)	maximum (minimum)
sup (inf)	supremum (infimum)
$\mathcal{R}$ and $\mathcal{C}$	field of real and complex numbers
$\mathcal{R}^n$ and $\mathcal{C}^n$	space of real and complex n-dimensional vectors
$\mathcal{R}^{n \times m}$ and $\mathcal{C}^{n \times m}$	space of n by m real and complex matrices
$\mathcal{H}_2\left(\mathcal{H}_2^n\right)$	signal space of all signals (n-dimensional vectors of signals) with bounded energy
$\mathcal{H}_\infty\left(\mathcal{H}_\infty^{n \times m}\right)$	function space of all transfer functions of stable systems (n by m-dimensional transfer function matrices of stable systems)
$\mathcal{RH}_\infty\left(\mathcal{RH}_\infty^{n \times m}\right)$	space of all rational transfer functions of stable systems (n by m-dimensional rational transfer function matrices)
$\mathcal{L}_\infty$	Lebesgue space of all functions essentially bounded on the imaginary axis

For the definitions of $\mathcal{H}_2$, $\mathcal{H}_\infty$, $\mathcal{RH}_\infty$ and $\mathcal{L}_\infty$, and the reader is referred to.

X^T	transpose of matrix X
X^*	conjugate transpose of (complex) matrix X
$X^\perp$	orthogonal complement of matrix X
X^{-1}	inverse of matrix X
X^+	pseudo-inverse of matrix X
X^-	left or right inverse of matrix X
$diag(X_1, \cdots, X_n)$	block diagonal matrix formed with $X_1, \cdots, X_n$
$X(i{:}j, p{:}q)$	submatrix consisting of the i-th to the j-th rows and the p-th to the q-th columns of matrx X
$col(X)$	vectorise X, $col(X) = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \in \mathcal{R}^{nm}$, for $X = [x_1 \cdots x_m] \in \mathcal{R}^{n \times m}, x_i \in \mathcal{R}^n, i = 1, \cdots, m$
$rank(X)$	rank of matrix X
$tr(X)$	trace of matrix X
$\det(X)$	determinant of matrix X
$\lambda(X)$	eigenvalue of matrix X
$\overline{\sigma}(X)$ $(\sigma_{\max}(X))$	largest (maximum) singular value of matrix X
$\underline{\sigma}(X)$ $\left(\sigma_{\min}(X)\right)$	least (minimum) singular value of matrix X
$\sigma_i(X)$	the i-th singular value of matrix X
$\Vert \cdot \Vert$	Euclidean norm of a vector
$\Vert \cdot \Vert_F$	Frobenius norm of a matrix
l_2-bounded	discrete-time signals with bounded energy
$\mathcal{L}_2$-bounded	continuous-time signals with bounded energy
$G(p)$	transfer matrix, p is either s for continuous-time systems or z for discrete-time systems
$G^*(j\omega), G^*\left(e^{j\theta}\right)$	conjugate of $G(j\omega), G\left(e^{j\theta}\right)$
(A,B,C,D)	shorthand for the state space representation
$rank\ (G(p))$	normal rank of $G(p)$
$\Vert G \Vert_2$	$\mathcal{H}_2$ norm of (stable) transfer function matrix G
$\Vert G \Vert_\infty$	$\mathcal{H}_\infty$ norm of (stable) transfer function matrix G

For the definitions of $\Vert G \Vert_2$ and $\Vert G \Vert_\infty$, the reader is referred to.

$\Pr(\mathrm{a} < b)$	probability that $a < b$
$\mathcal{N}(a, \sum)$	Gaussian distribution with mean vector a and covariance matrix $\sum$
$x \sim (a, \sum)$	x is distributed as $(a, \sum)$
$\chi^2(m)$	chi-squared distribution with m degrees of freedom
$\mathcal{E}(x)$	expectation of x
$var(x)$	variance of x
$cov(x,y)$	covariance (matrix) of x and y
$var(x) = cov(x)$	covariance (variance-covariance) matrix of vector x
i. i. d.	independent and identically distributed

Part I
Introduction, Basic Concepts and Preliminaries

Chapter 1
Introduction

The past two decades have witnessed tremendous development in the field of fault diagnosis and fault-tolerant control. This trend is the logic consequence of the ever increasing demands for highly economic and ecological operations of technical systems, processes and assets in all industrial sectors. The speed, at which new concepts, schemes and methods are developed, is rapid. In the course of this development, innumerable methods and successful applications have been reported.

Fault diagnosis technique is an engineering thematic area and its applications can be found across all technical fields. Consequently, in its history, the development of fault diagnosis technique and methods was obviously formed by individual technical demands and characteristics of systems and assets under monitoring. There was no uniform and standardised theory and framework. The situation has changed dramatically since this decade. Like all technical and scientific disciplines, in the era of information, digitalisation and big-data, the impact of artificial intelligence, computer science, information theory and communication technology on the development of fault diagnosis technique is enormous and everywhere. A uniform technological framework towards intelligent and data-driven fault diagnosis is being newly established.

Generally, fault-tolerant control (FTC) deals with feedback control systems. From the methodological viewpoint, fault-tolerant control is a thematic field of control theory and engineering. Logically, the development of FTC technique can be well characterised by the application of advanced control theoretical methods and associated mathematical tools. Robust control theory, adaptive control algorithms and newly optimisation methods including model predictive control (MPC) are the major methodological tools for the design and implementation of fault-tolerant control systems. The impact of the main technologies in the era of information and big-data on the FTC technique is reflected by the recent research efforts on FTC in networked control systems (NCSs) and cyber-physical systems (CPSs).

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_1

1.1 Trends and Mainstream in Research

In the current decade, the following new trends can be observed in comparison with the research efforts in the past decades,

- data-driven, multivariate analysis (MVA) and machine learning based methods are dominate in the field of process monitoring and fault diagnosis,
- research efforts on model-based fault diagnosis pounce on fault detection and estimation methods for special classes of systems like networked and distributed systems, event-triggered or switched systems,
- detection of two special types of faults, the so-called intermittent and incipient faults, is receiving remarkable attention, and
- in the research area of fault-tolerant control, efforts are strongly focused on
  - fault estimation and, based on it, fault compensation strategy, as well as
  - application of real-time optimisation techniques like MPC technique.

1.1.1 Data-Driven, Statistic and Machine Learning Based Fault Diagnosis Methods

The enthusiasm for machine learning (ML) and big data technologies significantly influences the developments in all engineering and scientific areas. The key issues in the diagnosis framework, like feature generation and analysis, (fault) classification and decision making, are also basic tasks in machine learning. It is a logic consequence that most of the existing ML methods and concepts have been introduced into the fault diagnosis framework, often in combination with MVA methods. At the very beginning of this development, the process of transferring an existing MVA-ML method to the fault diagnosis application was built with sophisticated research efforts, which resulted in certain time lag, but allowed sufficient time for necessary validations. In the course of this development, the time lag of the transfer processes has become shorter and shorter. Recently, it seems that it is becoming a competition of publishing applications of newly developed ML methods and algorithms to fault diagnosis. The most recent example is the application with deep learning technique. The consequence of this copy-and-paste style of research is that it is hard to have a solid overview of all published MVA-ML based fault diagnosis methods. Below, we would like to shortly review and analyse the basic ideas and common working principles of the existing MVA-ML based fault diagnosis methods without addressing the methods in detail.

Which MVA-ML methods are really useful for a reliable and efficient fault diagnosis? To answer this question, let us recall the basic tasks of a fault detection system as an example. Note that fault detection is the first and most importance task of fault diagnosis, and most of the fault isolation and classification problems can be re-formulated as a number of fault detection problems. A fault detection system or scheme is composed of three essential components and steps:

- data collection and pre-processing,
- feature generation and extraction, and
- decision making and classification algorithms.

Most of the published MVA-ML based fault diagnosis methods and algorithms are dedicated to the second and third steps on the assumption that the collected data have been well pre-processed and satisfy the required conditions. The basic idea behind the step with feature generation and extraction is to solve the core problem in fault detection: faults vs. uncertainties. Without the existence of uncertainties, fault detection is straightforward. Regrettably, in most of technical systems and assets, uncertainties are inevitable and exist in different forms. Measurement and process noises are the simplest form of uncertainties, which can be, for example, modelled as random variables with certain distributions. Variations in system and asset parameters caused by, for instance, varying system operation conditions, ageing in machine and asset components or changing environmental conditions around the system and asset, result in uncertainties which are hard to be described analytically. A feature is a function of all reliable (possibly pre-processed) measurement variables. It should be formed (generated) in such a way that it depends on the faults to be detected. Unfortunately, it is unavoidable that a feature is, more or less, corrupted with uncertainties. For the fault detection purpose, a good feature is a function that is sensitive to the faults and less affected by the uncertainties. In the step of decision making, a decision is made based on an analysis of the feature. The mostly adopted strategy is to compare the feature values in the fault-free and faulty operations, and a decision is made on account of the difference between these values. This requires that the feature (as a function) should be a metric or a good measure or an indicator for the influences of the faults and uncertainties. In an abstract form, we can summarise the fault detection problem as finding a metric for assessing the measurement data. Roughly speaking, a defined fault detection problem is optimally solved, when the ratio,

$$\frac{\text{the metric value corresponding to the faulty operation}}{\text{the metric value corresponding to the fault-free operation}},$$

is at the largest.

It should be emphasised that both data-driven and model-based fault detection schemes follow the above-described strategy. Their differences lie in the realisation and implementation. In the data-driven fault diagnosis framework, there are two types of strategies for the steps of features generation and decision making:

- modelling of a designed feature by learning,
- automated learning of the two steps in a single learning process.

In the first strategy, on account of certain assumptions or *a priori* knowledge of the system under consideration or applying some statistic and mathematical methods, a model mapping the process measurement variables to the feature is first established. By means of collected operation data, the model parameters and even the model structure are then identified. This process is called training or learning, and runs typically offline. Based on such a model, an online detection of faults can be realised by checking and analysing the feature value delivered by the model. Below are some examples. For the technical details, the reader is referred to the study in the subsequent chapters.

Example 1.1 *PCA based fault detection (referred to Sub-Sec. 3.5.4). Principal component analysis (PCA) is a simple fault detection method, in which two features, the so-called T^2 and SPE (squared prediction error) test statistics, are defined. Regrettably, in the literature the PCA method is generally introduced as a projection algorithm without describing (i) the assumptions for its use in fault detection and (ii) its statistic interpretations. In order to apply T^2 and SPE test statistics successfully to fault detection, (i) the process measurement variables are assumed to be normally distributed, (ii) a data pre-processing is needed, and (iii) the faults to be detected can be modelled as the so-called additive faults which cause changes in the mean (vector) of the process measurement variables. Moreover, T^2 test statistic is the so-called Mahalanobis distance which is a dissimilarity measure between two random vectors of the same distribution with the same covariance matrix. Finally, the detection logic is established based on the statistic distributions of both features. The model parameters of the T^2 and SPE test statistics are the mean (vector) and the covariance matrix of the process measurement variables, which should be estimated (learned) using the collected process data.*

Example 1.2 *KL divergence based fault detection (referred to Sect. 15.2). Kullback-Leibler (KL) divergence is a dissimilarity measure between distributions. When it is assumed that the fault to be detected would cause changes in the distribution of the measurement variables, KL divergence is a good feature for detecting such a fault. Since for the computation of the KL divergence model the distribution of the (random) measurement variables during the fault-free operation is needed, in the data-driven framework an estimation of the distribution is to be performed during the training phase using the collected process data. To this end, there exist numerous algorithms, among them the so-called kernel density estimation (KDE) method is very popular.*

Example 1.3 *SVM based fault detection (referred to Sub-section 18.3.2). Support vector machine (SVM) is a popular ML method. One of its application fields is classification, which can be directly adopted for the use of fault detection. In order to solve the problem of measuring the distance between two data sets, the faulty and fault-free data sets, a (threshold) hyperplane is introduced, which separates the faulty data set from the fault-free one, and thus guarantees a defined (Euclidean) distance between these two data sets. During the training, the hyperplane model is to be determined.*

Artificial neural networks (ANNs) are the most popular technique to realise feature generation and decision making in a single step. For example, for the fault detection purpose, an ANN is trained using the collected operation data in such a way that the ANN directly delivers the decision result for faulty (alarm) or fault-free. That is, an ANN-based fault diagnosis system is a mathematical model that describes relations between the measurement variables and the faults. During the training, this model is identified using the measurement data. In other words, training an ANN is in fact to perform a model identification. In application, the model is driven by the system measurement variables (as the input) and delivers information about the possible faults (as output).

The unbelievable media effects of recent remarkable successes of deep learning technique in some technical fields have initiated the enthusiasm for applying different forms of ANN schemes, in particular the deep learning technique, to fault diagnosis. Thanks to their self-learning capacity, ANNs are able to extract features of the faults to be detected from the training data automatically. And these features enable us to distinguish the faults from the uncertainties in the measurement data efficiently. On the other hand, its extreme dependence on the training data also limits the application of the ANN technique in fault diagnosis. One obvious problem is the availability of the so-called labelled data. Labelled data are data which have been recorded under known operation conditions. In order to train an ANN, the expected output of the model (to be identified) with respect to the given training data (as input) should be known. In fault detection applications, this means, whether the data have been recorded by faulty operation and thus labelled by faulty or by fault-free operations and labelled by fault-free should be known. In a reliable technical system, faulty operation is an event with (very) low probability, while most of the data are collected during fault-free operations (and thus labelled as fault-free). As a result, the number of the data samples for the faulty operation is in general much smaller than the number of the data labelled by fault-free. A further concern is the transferability. For instance, an ANN-based fault diagnosis system for an industrial asset has been well trained and constructed using a huge number of data collected as the asset is in operation. Even for all these efforts at great expense, it cannot be guaranteed that this diagnosis system can be used for another asset of the same type but located in a different operation environment.

A potential and promising solution to the above problems is the application of the so-called transfer learning technique. Transfer learning is a research field in machine learning and deals with, roughly speaking, development of methods which enable to apply learned knowledge of the solution of a defined problem to solving different but relevant problems. Recall that data-driven fault diagnosis can be schematically defined as a model identification problem and solved using the system data (as model inputs) and labels (knowledge) of the faults (as model outputs). To achieve fault diagnosis of two different systems, two different models are to be identified using two different system data sets and two labelled data sets. Assume that

- two systems, say S1 and S2, under consideration are similar,
- for one system, say S1, the operation data are well labelled, and thus its (diagnosis) model M1 is well identified,
- for system S2, there exist few data and they are poorly labelled.

Consequently, it is hard to identify a reliable diagnosis model M2 for system S2 using the available data. This problem can be addressed using transfer learning methods, which help us to use the well-labelled operation data and model of system S1 to improve the identification of model M2 based on probabilistic relations and using optimisation algorithms.

Example 1.4 *Application of transfer learning to machine diagnosis. It is well known that machine diagnosis becomes a challenging task, once a machine is integrated in a large-scale and complex system. Differently, under laboratory conditions, the same machine can be well tested and sufficient operation data can be collected. This allows us to build a reliable machine diagnosis model for the machine (under laboratory conditions). Now, transfer learning methods may support system engineers to apply machine diagnosis knowledge, including the machine diagnosis model and the operation data under laboratory conditions to diagnosing the machine integrated in the large-scale system. In the literature given at the end of this chapter, successful cases of this application have been reported.*

In their recent survey report (see the reference given at the end of this chapter), Lei et al. have pointed out that the transfer learning technique would be the major technological tool for data-driven fault diagnosis of the next decade.

1.1.2 Model-Based Fault Diagnosis Research

Model-based fault diagnosis techniques played once a very dominant role in the research domain of fault diagnosis. Even today in the era of information and digitalisation, model-based fault diagnosis methods are widely accepted as an efficient and powerful technique in dealing with fault diagnosis issues for dynamic systems.

After a dynamic development in the 1980s and 1990s, in which the framework for the model-based fault diagnosis techniques was established with three main research areas,

- observer-based fault detection, isolation and estimation,
- parity-space based fault detection and isolation, and
- parameter identification based fault detection and estimation,

all the three areas of the model-based fault diagnosis technique were well equipped with basic concepts, methods and algorithms. It is remarkable that most of these methods and algorithms are the well-established results in control theory with slight modifications. In the following decade, the research focus was mainly on robustness issues, which, due to the use of mathematical models, is a natural and necessary key step. In the course of this development, the foundations for system design techniques with the associated (mathematical) tools, for instance, $\mathcal{H}_\infty$ robust theory, handling of special classes of nonlinear systems like systems satisfying Lipschitz conditions, Takagi-Sugeno (T-S) fuzzy systems etc., system adaptive technique, sliding mode control methods, have been laid for the research efforts in the recent decade. In a certain sense, the most valuable contributions in this time period have been the "translation and standardisation works", which have resulted in formalisation of common fault diagnosis problems by means of problem formulations known in control theory and engineering. A representative example is the so-called $\mathcal{H}_-/\mathcal{H}_\infty$ design scheme for observer-based fault detection systems with LMI (linear matrix inequality) algorithms as the solution tool. It is noteworthy that the common basis of these techniques is Lyapunov theory. Reviewing the research results published in the current decade, it can be recognised that

- there are no significant contributions (i) to the model-based fault diagnosis framework, and (ii) to the existing essential design schemes, as established and developed in the past three decades,
- the main research efforts have been focused on addressing fault diagnosis issues, on the basis of the well-established formalisation framework, for special kinds of dynamic systems like systems with different types of time-delays, NCSs, distributed large-scale systems, switching systems, and newly multi-agent systems and CPSs, and
- in this regard, a great number of publications have been dedicated to the design algorithms with skilled application of existing mathematical and control theoretical tools.

Recently, a further trend can be observed. More and more reported research efforts have been devoted to the (robust) fault estimation issues with the argument that, once a fault is estimated, fault detection and isolation problems are solved as well. The real reason behind this handling is in fact a simplification of the problem formulation and handling. As mentioned, a fault detection problem is, in its core, a trade-off between the sensitivity to the fault and robustness against uncertainties or simply fault detection rate vs. false alarm rate. From the mathematical point of view, this is a multi-objective optimisation problem, and its solution is often a challenging task. In the framework of (robust) fault estimation, the problem is generally formulated as minimising the estimation error of the fault with respect to uncertainties. It is well-known in control theory that such a problem can be efficiently addressed in the framework of Lyapunov theory.

As a summary, it can be concluded that the research on the major model-based fault diagnosis techniques has been strongly driven and formed by the development of control theory. And, this trend will be reinforced rather than weakened.

1.1.3 Detection of Intermittent and Incipient Faults

Driven by industrial applications and continuously increasing demands for reliability and safety, recent research efforts have been devoted to two special types of faults, the so-called intermittent and incipient faults. Intermittent faults are malfunctions in technical systems and assets, which occur from time to time. Recall that most of the existing fault diagnosis methods have been developed on the assumption that the fault under consideration persists after its occurrence. In practice, it can be, on the other hand, often observed that a fault occurs in the system only for a (short) time interval, then disappears. And, this scenario is repeated until the component or the whole system fails. Typical examples are

- faulty rotary components in mechanical assets, which are, for instance, mounted in wind turbines, or
- faulty components of an electrical or electronic circuit embedded in a large-scale system, or
- software faults, which only occur under certain logic conditions.

Nowadays, industrial systems become very complex. Full integration of hardware and software, mechanical and electronic components and sub-systems into industrial systems is the technical state of the art. This trend calls for novel methods for efficiently detecting intermittent faults. Corresponding to this development, considerable research efforts have bee reported, often well supported with research funding.

In the course of industry 4.0, condition monitoring (CM), prognostics and health management (PHM) become key technologies aiming at meeting high industrial reliability and safety requirements. In the CM-PHM framework, research in the thematic field of predicting and detecting incipient faults becomes exceptionally active. An incipient fault is in fact a transient process in a system component or in an asset. It is the beginning of a degradation process in the functionality and in the operating performance of the component or the asset. Incipient faults are generally characterised by their low magnitude and transient behavious. These properties make detection of incipient faults very tricky. On the other hand, detecting or even predicting incipient faults are a key pre-request for a successful PHM. This strongly drives the recent research activities on detection and prediction of incipient faults.

From the technological point of view, the research on diagnosis of intermittent and incipient faults is still in the initial stage. Most of the reported research efforts have been devoted to the application of the existing fault diagnosis methods with moderate modifications.

1.1.4 Fault-Tolerant Control

Similar to the model-based fault diagnosis technique, the development of the fault-tolerant control technique in the past decade has been strongly driven by the advances in control theory and engineering. During the initial phase beginning in the 1990s, basic FTC strategies, concepts and system design schemes were well developed and, as a result, a fundamental framework was established. The recent research activities mainly concentrate on implementing the existing FTC strategies using advanced control theoretical methods. In this course, the following trends can be identified:

- fault estimation based FTC schemes become very popular, and
- advanced methods of real-time optimisation are widely applied for achieving FTC.

As mentioned before, thanks to its general form of the problem formulation that well fits into the framework of Lyapunov theory, robust fault estimation is also a main research focus in the fault diagnosis area. An extension to fault-tolerant control is straightforward and can be realised, for example, by feeding back the estimated fault aiming at compensating the faulty effects in the control system under supervision. The design of such a compensator is generally performed in the framework of Lyapunov stability theory and, in most cases, done offline. The major advantage of this strategy is that it can be applied to different types of dynamic systems. Consequently, major efforts are often mathematically skilled application of certain existing control theoretical methods to a common fault-tolerant control problem for a class of control systems.

In recent years, the control community has received a strong boost from the considerable development of real-time optimisation techniques and algorithms. MPC and adaptive control techniques become much more powerful and efficient in implementing online adaptation and tuning of feedback controllers. In combination with diagnostic algorithms, these methods can be applied to realising fault-tolerant control. It is of remarkable interest to notice that such fault-tolerant control algorithms result in optimal system performance under the (faulty) operation conditions. This is the major advantage over the traditional fault-tolerant control schemes.

1.2 Motivation

1.2.1 Data-Driven and Model-Based Fault Diagnosis

The triumphal march of the statistical and machine learning based fault diagnosis techniques has also its downside. Many researchers fall in love with ML methods, concentrate on transferring (newly developed) ML methods to fault diagnosis under some vague formulation of fault diagnosis problems, but pay less attention to the specifications of fault diagnosis in technical systems. It is not rare that a problem has been interpreted as an important issue in fault diagnosis, although it is of less practical interest and importance in that regard, only because the applied ML method has been initially developed for dealing with such a problem. A further phenomenon is incorrect applications of well-established and popular ML methods to dealing with fault detection problems. Here are two unexceptional examples.

Example 1.5 *PCA technique is a representative and very popular tool to transform (map) the data to a (much) lower dimensional subspace, the so-called principal component subspace (PCS), and simultaneously attempts to preserve the information in the data. Depending on applications, the information here is related to some features of interest, for example, image features in image processing. Consequently,* *it is often the case that strong attention is paid to the PCS by applying the PCA technique to fault detection. As we know (see also the discussions in Chap. 3 and 13), in the fault detection framework, the PCA algorithm is performed on the basis of covariance matrix of the measurement variables, which represents the uncertainty in the measurement process. In other words, the PCS preserves the dominant uncertainty in the data. As a result, a fault detection performed in the principal component subspace results in poorer detectability. By the way, it is well-known in ML framework that PCA is an unsupervised learning algorithm. In its application to fault detection, the PCA algorithm is generally implemented using fault-free data. Thus, the PCA learning is done using the labelled data and supervised.*

Example 1.6 *Kernel PCA (KPCA) is another popular and representative method adopted to deal with fault detection issues. The core of this method is to transform the data to a higher-dimensional subspace by means of a nonlinear function. The theoretical foundation for such (nonlinear) data transforms is the Cover's theorem, which claims the existence of some nonlinear functions. These functions transform the data, which are not linearly separable, to a higher-dimensional subspace, in which the transformed data become, with high probability, linearly separable. Having transformed the data, the standard PCA algorithm can be then effectively applied to solving the fault detection problem. To this end, the so-called T^2 test statistic used in the PCA algorithm is also applied for the online fault detection. Without knowing the statistical meaning of the T^2 test statistic, this way of handling sounds reasonable and logic. The reality could be different. It is well-known in statistics that the T^2 test statistic is the so-called Mahalanobis distance that is a dissimilarity measure between two random vectors of the identical distribution with the same covariance matrix. In other words, the Mahalanobis distance is used for checking the deviation of a measurement point (data) from the given mean (center). In this context, it is very questionable to apply the T^2 test statistic for the detection purpose after the data transformation, since (i) for the training only fault-free data are used, and (ii) nonlinear mappings generally do not guarantee to preserve the statistical properties. In order to illustrate this point, we extend a popular (academic) example adopted in the literature on KPCA studies (see the references given at the end of this chapter) by including additive faults. Given two measurement (random) variables (x, y) with*

$$x = \begin{cases} \varepsilon_x \sim \mathcal{U}(-1, 1), \text{ fault-free}, \\ f_x + \varepsilon_x, \; f_x \neq 0, \text{ faulty}, \end{cases}$$

$$y = \begin{cases} cx^2 + \varepsilon_y, \varepsilon_y \sim \mathcal{N}\left(0, \sigma^2\right), \text{ fault-free}, \\ cx^2 + \varepsilon_y + f_y, \; f_y \neq 0, \text{ faulty}, \end{cases}$$

where $c > 0$ is some constant, f_x, f_y represent (deterministic) sensor faults and $\varepsilon_x, \varepsilon_y$ are independent. Now, we apply the kernel method, as suggested in the references. Suppose that N data have been collected, $(x_i, y_i), i = 1, \cdots, N$. Let the kernel function be

$$k(z_i, z_j) = e^{-\frac{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}{2}}, z_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}, k = i, j.$$

It results in the so-called kernel matrix K with the (i, j)-th entrance

$$k_{ij}\left(z_i, z_j\right) = e^{-\frac{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}{2}}, i, j = 1, \cdots, N,$$

based on which the principal components will be determined. Note that

$$k_{ij}\left(z_i, z_j\right) = \begin{cases} e^{-\frac{\left(\varepsilon_{x_i} - \varepsilon_{x_j}\right)^2 + \left(c\varepsilon_{x_i}^2 - c\varepsilon_{x_j}^2 + \varepsilon_{y_i} - \varepsilon_{y_j}\right)^2}{2}}, \textit{ fault-free,} \\ e^{-\frac{\left(\varepsilon_{x_i} - \varepsilon_{xj}\right)^2 + \left(c\left(\varepsilon_{x_i} + f_x\right)^2 - c\left(\varepsilon_{x_j} + f_x\right)^2 + \varepsilon_{y_i} - \varepsilon_{y_j}\right)^2}{2}}, \textit{ faulty,} \end{cases}$$

where

$$\varepsilon_{x_i}, \varepsilon_{x_j} \sim \mathcal{U}\left(-1, 1\right), \varepsilon_{y_i}, \varepsilon_{y_j} \sim \mathcal{N}\left(0, \sigma^2\right), i, j = 1, \cdots, N,$$

are samples of ε_x, ε_y, respectively. It is obvious that the kernel matrices in the fault-free and faulty cases are different. Thus, it is not suitable to apply the Mahalanobis distance as the test statistic. In summary, the use of T^2 test statistic and, based on it, the corresponding threshold setting are statistically unfounded.

Further similar examples can also be observed from the application of popular manifold learning algorithms like locally-linear embedding (LLE) and Isomap methods, which were developed for nonlinear dimensionality reduction, to fault detection. The reported studies often follow the same pattern:

- mapping the (high-dimensional) measurement data by means of a manifold learning algorithm to a low-dimensional data subspace, and
- applying the existing test statistics or methods to the low-dimensional data to achieve fault detection.

It is obvious that the major deficits in the course of applying the well-established ML-methods to dealing with fault diagnosis issues are:

- less attention has been paid to the core task of fault detection: find a test statistic or evaluation function, on which the influence of uncertainties (in the fault-free data) is minimised and, *simultaneously,* the impact of the fault (to be detected) is maximised,
- a hard combination of the ML algorithms and the statistical decision (test statistics and threshold setting) methods under the motto

$$\begin{aligned} &\text{an ML algorithm} + \text{a statistical decision method} \\ &= \text{a new fault detection method,} \end{aligned}$$

  is often not targeted and adequate, and
- requirements and specifications of fault diagnosis are often different from the original application and development objectives of the adopted ML methods.

One deeper reason for this situation could be the missing methodological basis of fault diagnosis framework in education and research. To our best knowledge, there exists no methodological paradigm, in which fault diagnosis problems are formulated in simple forms, but applicable for most types of technical systems. Here, we would like to emphasise the fact that data-driven and model-based fault diagnosis methods address, from the fault diagnosis viewpoint, the same complex of fault diagnosis problems. In this context, they should share the same methodological paradigm with the same problem formulations. Unfortunately, they are generally dealt separately and in different regards due to the missing methodological basis. It is a motivation of our work to make contributions to change this situation.

It is a common opinion that model-based methods are more powerful in dealing with fault diagnosis in dynamic processes, in particular in automatic control systems, than data-driven schemes. On the other hand, our observation of state of the art of the publications in this field, as reviewed in the previous sub-section, reveals that

- fault diagnosis problems are generally addressed on the basis of the formalisation in control theory and engineering framework and less fault diagnosis specifically,
- model-based methods have been rarely devoted to dealing with fault diagnosis issues in the probabilistic framework,
- the recent research works are control theoretical method driven and follow closely the new and popular topics in control theory, and
- less attention has been dedicated to the investigation on the fault diagnosis oriented and specified methods. Less research efforts have been made to study fault diagnosis issues for basic but general classes of dynamic systems like time-varying systems (which cover a wide spectrum of system types popularly handled recently) or nonlinear systems (not special types of nonlinear systems).

Fault detection in feedback control systems is a special topic in the thematic area of model-based fault diagnosis. It is strongly related to the study on fault-tolerant control and is of considerable practical importance in process and automotive industries, in robotics and mechatronic systems, just mentioning some representative industrial sectors. Also in this research field, the application of the existing control and model-based diagnosis methods is the state of the art. Often, the developed methods are dedicated to detecting and estimating faults in the typical components embedded in a control loop like sensors and actuators.

On account of the above observations and analysis, we are well motivated to

- establish a methodological paradigm that can be shared by both data-driven and model-based frameworks and gives a generalised formulation of basic fault diagnosis issues,
- study basic fault diagnosis problems for general classes of dynamic systems including linear time-invariant (LTI), linear time-varying (LTV) and nonlinear systems,
- analyse existing basic data-driven fault diagnosis methods with a strong focus on their statistic properties, and to propose, where needed, alternative schemes,

- apply the solutions of the generalised fault diagnosis problems to dealing with fault diagnosis in distributed and networked large-scale processes, which characterise automatic control systems of the next generation,
- develop a scheme, by which the model-based fault diagnosis issues can be addressed in the probabilistic framework as well, and
- find alternative and practice oriented ways to handle fault diagnosis issues in feedback control loops.

In our work, we will pay attention to incipient faults from the following aspects:

- checking, when possible, how far a fault diagnosis method is applicable for detecting incipient faults,
- developing performance degradation prediction methods which enable an efficient prediction of performance degradation caused by incipient faults, and
- establishing a framework for recovering the system performance from the degradation.

1.2.2 Fault-Tolerant Control and Performance Degradation Recovery

As described before, the major focus of the existing fault-tolerant control schemes is on compensating and accommodating influences of faulty system components like sensors, actuators or some other hardware units. The major objective is to fully or partially recover the functionality of those faulty components so that the overall system performance degradation is limited. In this regard, these fault-tolerant control schemes can be viewed as component oriented. On the other hand, it can be observed that in the course of industry 4.0, considerable efforts have been made in the recent decade to increase the component reliability and, more recently, to increase the intelligent degree of those key system components. Smart sensors and actuators are nowadays the state of the art. The new generation of smart system components will be characterised by their ability of self-diagnosing and self-repairing.

Aiming at following the trends in automation industry and shaping our research to meet the industrial and application demands, we propose and study the so-called performance-based fault-tolerant control as an alternative strategy to the component oriented technique. Our focus lies on recovering system performance from the performance degradation caused by faults or unexpected changes in operation conditions or even mismatching among controller parameter settings. The methods to be developed should satisfy the following requirements on

- plug and play (PnP) implementation,
- real-time applicability and
- online adaptation and learning.

PnP is a challenging topic in the field of fault-tolerant control. PnP implementation enables (i) to embed additional controllers without changing the existing controllers towards recovering the performance, or (ii) to run certain maintenance or repair actions online. Online learning and simultaneously satisfying real-time requirements will be a novel feature of fault-tolerant control systems of the next generation. Our objective is to contribute to this development by transferring machine learning methods to the fault-tolerant control area.

In our work on the performance-based fault-tolerant control, both structural system performance like the system stability as well as control performance expressed in terms of certain cost functions are under consideration. This work is closely related to the intended efforts on detection and prediction of performance degradation. In fact, performance degradation monitoring and recovery are two key steps of our performance-based fault-tolerant control strategy.

1.2.3 Performance Assessment of Fault Diagnosis and Fault-Tolerant control Systems

It is popular practice in the fault diagnosis and fault-tolerant control research domain that a published method, its efficiency and performance are demonstrated by a simulation or case study. In more elaborate studies, benchmark (case) study is adopted to perform comparison with other related methods and thus show convincingly that the proposed method is an improvement of the state of the art technique. This way of demonstrating the capability of the published results is very popular in the data-driven research area. For the following reasons, this manner of performance assessment is critical.

As mentioned, the core of fault diagnosis is suitable trade-off handling of faults and uncertainties. Both uncertainties and faults are in their nature random variables. This demands for a fair performance assessment of fault diagnosis and fault-tolerant control systems in the probabilistic and statistic framework. Due to the limitation of simulation capacity and availability of the data amount, results from a benchmark study are generally less representative and of lower statistical significance. In addition, although those performance indices like false alarm rate and fault detection rate are well defined in the probabilistic framework, the computation rules are rarely followed in most of published benchmark or case studies.

Triggered by these concerns, we will propose a probabilistic framework and related technique aiming at

- developing computation algorithms of fault diagnosis performance indices,
- using them to achieve fair performance assessment of fault diagnosis systems, and based on them,
- developing methods for designing fault diagnosis and fault-tolerant control systems.

1.3 Outline of the Contents

This book is composed of six parts. Besides Part I, which serves as a common basis for the subsequent studies, Parts II–VI are dedicated to five different thematic areas. They are self-contained and so structured that they can also be used for self-study on the concerned topics.

1.3.1 Part I: Introduction, Basic Concepts and Preliminaries

The objective of this part is to (i) introduce definitions of basic fault detection and estimation problems, which are independent of system types and methods applied for achieving fault diagnosis, and (ii) to review known definitions, concepts and existing fault diagnosis schemes, which as preliminaries are necessary for our subsequent work.

Recall our intention to establish a methodological paradigm being able to be shared by both data-driven and model-based frameworks and to give a generalised formulation of basic fault diagnosis issues. Chap. 2 serves for this purpose. Fault detection and estimation problems are first formulated and described in an abstract form. Among these defined fault diagnosis problems, *fault detection with maximum fault detectability* is the optimal fault detection problem mostly addressed in the subsequent chapters. This enables the formulation of basic optimal fault detection and estimation problems, which are applicable for most of system classes. Besides, at this abstract level, general solutions for the formulated optimal fault detection and estimation problems are provided. They also serve as guidelines for the subsequent investigations on the concrete solutions for different types of systems.

Chap. 3 is dedicated to (i) the review of basic methods for fault detection and estimation in static processes, and (ii) highlighting their solutions in the context of the optimal fault detection and estimation problems formulated in Chap. 2. The issues of fault detection and estimation in (static) processes either with noises or with deterministic disturbances are addressed both in the model-based and data-driven fashions. It is worth remarking that rare research results on fault diagnosis in static processes with deterministic disturbances have been reported in the literature.

Chap. 4 consists of (i) a review of basic model-based schemes for residual generation in LTI systems, in which the observer-based and parity space based methods are presented, (ii) two optimal model-based schemes for detecting (additive) faults in LTI systems with stochastic and deterministic unknown inputs, and (iii) the data-driven fault detection schemes for LTI systems. These two optimal model-based schemes are namely the Kalman filter based scheme for the stochastic case and the so-called unified solution ($\mathcal{H}_2$ observer-based) for the deterministic case. In this regard, it is further demonstrated that these two schemes solve the optimal fault detection problems formulated in Chap. 2. Besides, the so-called system factorisation technique, including left and right coprime factorisations as well as co-inner-outer factorisation, and associated with it, the stable kernel representation (SKR) model form are introduced. They are important mathematical tools intensively applied in the subsequent chapters.

Fault diagnosis in feedback control systems and fault-tolerant control are the major topics of this book. Needed preliminary knowledge for our study is introduced in Chap. 5. This includes, (i) the so-called Youla parameterisation of all stabilising controllers, (ii) various realisation forms of the Youla parameterisation, and (iii) the so-called fault-tolerant control architecture that has been developed based on the observer-based realisation of Youla parameterised controllers. In addition, the dual form of the SKR, the so-called stable image representation (SIR) is presented. It should be emphasised that the residual signal plays a central role in all these control schemes. In fact, the use of residual signals enables a deeper study on control, observation and detection issues from the information point of view. This is also the common thread through this book.

1.3.2 Part II: Fault Detection, Isolation and Estimation in Linear dynamic Systems

Although it is the common opinion that great research efforts in the past three decades have resulted in a solid framework for dealing with fault diagnosis issues in linear dynamic systems, a number of problems are still open. Among them are, for instance, optimal fault detection and estimation in time-varying systems, detection and isolation of multiplicative faults in LTI systems with uncertainties. The objective of this part is to investigate potential solutions of these open problems.

In Chap. 6, the unified solution is extended to a more general case. This extension allows us to design an optimal fault detection filter (FDF) in the case that the fault vector builds a lower dimensional subspace in the measurement space. To this end, two design schemes are proposed. The first one is an algebraic solution based on the (algebraic) system input-output model. The involved computations in the system design are lower and, in particular, straightforward. This fault detection scheme can be applied to both stochastic and deterministic systems. Also a data-driven implementation form of this scheme is derived. The second scheme is developed using a published algorithm for the co-inner-outer factorisation of LTI systems satisfying the above-mentioned structural restriction. The design algorithm is elegant, but demanding and sophisticated. It is illustrated that this solution solves the optimal fault detection problems formulated in Chap. 2 for the given class of LTI systems.

The main objective of Chap. 7 is to study fault detection issues in linear discrete-time varying (LDTV) systems. The focus is on deriving an optimal solution for the fault detection problem defined in Chap. 2. To this end, various mathematical tools are applied, which result in two solutions, one is based on the algebraic model and the other is derived using operator theory. Both solutions lead to the identical setting for the LDTV fault detection filter that can be viewed as an extension of the LTI unified solution. Aiming at gaining deeper insights into the solutions, the achieved results are further investigated in the context of co-inner-outer factorisation of LDTV systems. Recall that for LTI systems, inner and co-inner are defined in terms of the transfer function matrix of the system under consideration, and a co-inner can be expressed as the transpose of an inner. From the viewpoint of energy balance, an inner system is lossless with respect to the defined (energy) supply rate. In order to define the co-inner-outer factorisation of LDTV systems for our purpose, we introduce the concept of lossless with respect to information transform rate, as a dual form of the lossless property in the regard of energy balance. It is demonstrated that the optimal LDTV solutions can be achieved using a co-inner-outer factorisation. Further studies are dedicated to the relations between the achieved solutions and some optimal indices based solutions. We would like to mention that the LDTV state space representation is a general model form of linear dynamic systems. Systems like switched systems, linear parameter varying (LPV) systems, networked systems or event-triggered systems can be well modelled as LDTV systems.

Fault estimation in dynamic systems is receiving considerable attention in the research field of fault diagnosis and fault-tolerant control. Chap. 8 is dedicated to the topic of fault estimation in LDTV systems. Different from the robust unknown input observer and augmented observer schemes, which build the mainstream in the research field of observer-based fault estimation, we investigate fault estimation from the *least squares* (LS) optimisation viewpoint. The so-called least squares observers are in fact the analogue form of the celebrated Kalman filter and can be applied to the estimation of state variables in processes with deterministic unknown inputs. The mathematical tool for the solution of our least squares estimation is the regularised least squares (RLS) estimation method. Two LS fault estimation problems are addressed in our work, one for estimating sensor type of faults and the other for process faults. In addition, the relations between the unified solution (with faults as unknown inputs) and the LS estimation algorithms are analysed.

While the previous chapters are mainly dedicated to the diagnosis issues in the regard of additive faults and without considering model uncertainties, Chap. 9 deals with detection and isolation of multiplicative faults in LTI systems with model uncertainties. Multiplicative faults, also those with small size, may cause remarkable changes in the system structure and dynamics. They often rise up in a continuing process, are thus hard to be detected, in particular, when these faults are embedded in a closed-loop control system. Detecting and isolating multiplicative faults are challenging and open issues that are of significant research and practical interests. Due to the complexity of the involved problems, studies in Chap. 9 are multifaceted and include the following issues:

- Considering that (model) uncertainties and multiplicative faults could be presented in a system in different forms, the modelling issue of different types of uncertainties and faults is first addressed. In this work, the SKR and SIR model forms are at the centre of the discussion. On account of the equivalent relations between the various forms of uncertainties and faults, in the subsequent studies, the so-called left coprime factor form is adopted for representing the model uncertainties and multiplicative faults.

- Observer-based fault detection schemes are proposed for systems in open-loop and closed-loop configurations respectively. These schemes consist of (i) an observer-based residual generator design algorithm, and (ii) a threshold setting rule. In case of a closed-loop feedback control system, the influence of the controller on the detection performance is analysed. It is revealed that increasing the stability margin of the control system by minimising the $\mathcal{H}_\infty$ norm of the controller SIR enhances simultaneously the fault detectability. It should be remarked that this result is in direct contradiction with the common conjecture that fault detectability would be weak in a well-working feedback control loop.
- Noticing that less attention has been paid to qualitative fault detection and isolation (FDI) performance evaluation, further efforts are devoted to the performance analysis of observer-based FDI systems. A quantisation of FDI performance is of considerable practical interest and helpful to get a deeper insight into the system structural properties and for establishing appropriate design objectives. To this end, the coprime factorisation and gap metric techniques are applied as the control theoretical tool. Gap metric technique is widely applied in robust control theory. Roughly speaking, a gap is a measurement of the distance between two closed subspaces in Hilbert space. For the application of FDI performance analysis, we introduce the definition of the so-called $\mathcal{K}$-gap and derive its computation algorithm. The $\mathcal{K}$-gap metric is the dual form of the gap metric. Applying the $\mathcal{K}$-gap and the associated $\mathcal{L}_2$-gap metric, the proposed observer-based fault detection systems are analysed. Moreover, fault detection performance indicators are defined in terms of the $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metric. And the concept of fault-to-uncertainty ratio (F2U) is introduced. These results provide us with valuable quantisation of FDI performance and is thus helpful for designing observer-based systems for a reliable detection of multiplicative faults in uncertain systems.
- At the end of this chapter, the isolation topic of multiplicative faults is addressed with the help of the $\mathcal{K}$-gap. Using the SKRs as the clustering models of fault patterns and the $\mathcal{K}$-gap as the distance measure of the fault clusters under consideration, isolability of multiplicative faults is first defined. On this basis, fault isolation problems are formulated and, corresponding to it, an observer-based fault isolation scheme as well as an SKR identification based fault isolation strategy are developed. The latter fault isolation strategy includes an online identification of the SKR of the faulty system and data-driven computation of the system $\mathcal{K}$-gap. It is suggested that these two fault isolation methods could be applied in combination. With this work and the achieved results, a framework for isolating multiplicative faults is established.

1.3.3 Part III: Fault Detection in Nonlinear Dynamic Systems

There is no question that nonlinear observer-based fault detection is the most challenging topic in the fault detection research area. In recent years, much attention has been paid to the application of techniques like fuzzy technique, LPV methods or sliding mode technique to addressing nonlinear fault detection issues. Besides, considerable research efforts concentrate on systems with a special class of nonlinearities, typically Lipschitz nonlinearity or sector bounded nonlinearity. It is a surprising observation that little attention has been paid to the existence conditions of nonlinear observer-based fault detection systems for a general type of nonlinear systems, although this is a fundamental issue for the design of any type of nonlinear systems. Also, the parameterisation and optimisation issues of nonlinear observer-based fault detection systems are rarely addressed. The objective of this part is to make contributions to the research on these topics.

In the first section of Chap. 10, existence conditions for a general type of nonlinear observer-based fault detection systems are investigated. This work is helpful to gain a deeper insight into the fundamental properties of nonlinear observer-based fault detection systems. Corresponding to the use of two different types of evaluation functions, $\mathcal{L}_\infty$ and $\mathcal{L}_2$ types of observer-based fault detection systems are first defined. Inspired by the framework of input-state, input-output stability (IOS) and stabilisation of nonlinear control systems, the existence conditions for these two types of observer-based fault detection systems are then derived. In the second section of this chapter, various design schemes for the proposed observer-based fault detection systems are developed for their use under different system conditions.

System parameterisation is essential for system analysis and optimisation. Motivated by the well-established parameterisation of LTI residual generators, Chap. 11 is devoted to the parameterisation of nonlinear observer-based fault detection systems. For our study, the factorisation technique, the nonlinear SKR and SIR and input-output operator techniques serve as the mathematical tool. It is demonstrated that observer-based residual generators can be parameterised in form of a cascade connection of a system kernel representation and a post-filter. Moreover, using IOS methods and the results achieved in Chap. 10, parameterisations of different types of nonlinear observer-based fault detection systems, including the residual evaluator and the threshold, are investigated.

In Chap. 12, optimal design of observer-based nonlinear fault detection systems is investigated in the context of the optimal fault detection problem formulated in Chap. 2. The study is restricted to a class of nonlinear systems, the so-called affine systems. The core of our work is a co-inner-outer factorisation of nonlinear affine systems, which leads to the optimal construction of an observer-based fault detection system, analogue to the design schemes for LTI and LDTV systems. For our purpose, preliminary about Hamiltonian systems is first introduced. Based on it, co-inner and further co-inner-outer factorisation are defined. Noticing that there are few existing results on co-inner and the available definition is not coincident with the lossless interpretation, we propose a co-inner definition using the concept of lossless with respect to information transform rate, as introduced for the co-inner of LDTV systems in Chap. 8. On this basis, we are able to solve the co-inner-outer factorisation by finding a canonical transformation, which is then solved using the so-called generating function approach well-established in Hamiltonian mechanics. Having achieved a co-inner-outer factorisation, the initial design problem can be completely solved by setting the threshold and constructing a post-filter as described in the last section of this chapter.

1.3.4 Part IV: Statistical and Data-Driven Fault Diagnosis Methods

This part includes three chapters. They all deal with statistical and data-driven fault diagnosis, but are dedicated to different issues and not strongly related.

Our observations of the current trends in the statistical and data-driven fault diagnosis, as described in the previous two section, motivates a critical review of basic methods in the framework of MVA based fault detection methods in Chap. 13. It should be emphasised that this is not a literature review. Instead, the objectives of the review are (i) to stress and correct popular but misleading use of some standard techniques or methods, (ii) to pose critical questions on some basic multivariate analysis based fault diagnosis methods, and (iii) to motivate development of alternative MVA based fault detection methods, which will then be in part addressed in Chap. 15. In this regard, the following issues are addressed.

- On projection technique and its use in fault detection: It is state of the art that in many data-driven methods, projecting or transforming process data from the measurement subspace to another subspace with a reduced dimension is adopted. PCA technique is a representative example. With the PCA method as a reference, the arguments for and against the use of this technique are discussed.
- Data centering, time-varying mean and variance: In most of MVA based fault detection methods, centering the raw process data is the first step both in the offline training and online detection. In many publications, this step is not mentioned explicitly and thus less attention has been paid to the problems that may arise. We discuss the two possible problems with (i) data centering as well as the associated conditions and consequence, and (ii) handling of a time-varying mean.
- On detecting multiplicative faults and the use of T^2-test statistic: In the MVA based fault detection framework, a multiplicative fault is referred to the changes in the covariance matrix. Although investigations on detecting multiplicative faults by means of different test statistics have been reported, T^2-test statistic is still the mostly used one, also for detecting multiplicative faults. In particular, if there is no specification for the type of the fault under consideration, T^2-test statistic is the standard choice. It is illustrated that miss detection rate would be high when the T^2-test statistic is applied to detecting multiplicative faults.

- Assessment of fault detection performance: Here, confidential computations of FAR and FDR are discussed.

Fault detection in large-scale, interconnected and distributed systems is a challenging issue. In the literature, the major focus in this research domain is on the design of distributed fault detection systems towards distributed online fault detection. Consequently, model-based methods are mainly applied. Our work in Chap. 14 is dedicated to data-driven fault detection issues in interconnected and distributed large-scale systems. That means, both offline learning and online fault detection are to be performed in a distributed fashion. Two different classes of large-scale processes are under consideration: (i) large-scale processes equipped with a distributed sensor (monitoring) network, and (ii) interconnected large-scale processes with weakly coupled sub-processes. For the first class of processes, detecting the faults within the process (as a whole) from the local sensor nodes is the objective of the study, while the objective of fault detection in the second class of processes consists in detecting faults in each sub-processes. From the optimal fault detection viewpoint, as formulated in Chap. 2, the work in this chapter is the application of the optimal fault detection solution to interconnected and distributed large-scale systems. Considering the special role of communication networks and their topology in system operations, we first briefly introduce preliminary knowledge of network and graph theory. For detecting faults in a large-scale process by means of a distributed sensor network, we propose to apply the well-developed average consensus technique for a fusion of process data received by the sensor network. Various average consensus based detection algorithms are developed for an optimal fault detection in static large-scale processes equipped with a distributed sensor network. Based on a lifting data-driven SKR model of large-scale dynamic processes, we propose a consensus Kalman filter based distributed fault detection scheme, in which the structure of the distributed Kalman filter reported in the literature is adopted.

Chap. 15 is the follow-up discussion of the questions concerning the test statistics used for decision making and the metric adopted in a test statistic for measuring the distance between two probabilistic distributions. The objective of this discussion is (i) to select right test statistics for the fault detection problem under consideration, and (ii) to propose alternative methods when the conditions for the use of the existing test statistics are not satisfied. To this end, a general formulation and solution of generalised likelihood ratio (GLR) based fault detection are first presented. Under certain conditions, the GLR based detection method delivers the optimal fault detection performance as defined in Chap. 2. As a well-established dissimilarity measure between two distributions, Kullback-Leibler (KL) divergence is also widely accepted as a test statistic for the fault detection purpose. In order to gain a deeper understanding, the statistic properties of KL divergence are closely examined. It is demonstrated that the KL divergence is the expectation of likelihood ratio (LR). Unfortunately, the property that KL divergence is asymmetric has not received reasonable attention by some existing KL divergence based fault detection algorithms, in which the KL divergence from the faulty distribution to the fault-free distribution is adopted. Relationships between the KL divergence and GLR based fault detection methods, as well as the asymptotic behaviour of GLR and KL divergence used as test statistics are further investigated. Under consideration that identifying a distribution to cover the overall (normal) operation of a process variable is generally a technical challenge, due to uncertainties within and around the process under consideration, we propose in the last section of this chapter a pure data-driven method whose application requires no statistic knowledge *a priori*. The idea behind this method is to handle the (measurement) data sets from the viewpoint of information geometry and to abstract the overall process operations as a manifold, and consequently, to apply differential-geometric theory to solving fault detection and isolation problems. To be specific, the collected data are formed as symmetric positive definite (SPD) matrices, which can then be presented as a Riemannian manifold. For our purpose, we first introduce very basic differential-geometric properties of SPD matrices as a Riemannian manifold as well as some relevant concepts and methods, including definitions and relations like tangent space, geodesic curves, exponential and logarithmic maps as well as Riemannian distance. On this basis, we propose (i) Riemannian distance based basic fault detection algorithms, and (ii) algorithms for clustering on Riemannian manifolds and their application to fault detection and diagnosis.

1.3.5 Part V: Application of Randomised Algorithms to Assessment and Design of Fault Diagnosis Systems

This part is motivated by the discussion in Sub-section 1.2.3 on the correct assessment of fault diagnosis performance and an attempt to give convincing answers to the questions raised in the discussion. The theoretical fundament for our work is probabilistic methods called Randomised Algorithms (RA). The objective of this part is to establish a probabilistic framework to deal with assessment and design issues of fault diagnosis systems. This framework consists of three functional levels, (i) probabilistic models for faults and uncertainties, (ii) performance assessment of fault diagnosis systems in terms of FAR, FDR and mean time to fault detection (MT2FD) as well as the associated computation algorithms, (iii) design of fault diagnosis systems in the context of trade-off between FAR and FDR.

Chap. 16 is devoted to the construction of the first functional level. To this end, probabilistic models for dynamic systems with various types of uncertainties as well as probabilistic fault and evaluation function models are defined and described. It is followed by an introduction to the preliminaries of randomised algorithms.

To complete the construction of the second functional level, assessment of fault detection performance and computation algorithms are addressed in Chap. 17. Based on the probabilistic models for the uncertainties and faults, concepts like false alarm rate (FAR) with respect to (w.r.t.) a uncertainty mode, average false alarm rate (AFAR), fault detection rate (FDR) w.r.t. a fault pattern, average fault detection rate (AFDR) and mean time to fault detection (MT2FD) w.r.t. a fault pattern are introduced for the assessment of fault detection performance. Aided by the RA technique, randomised algorithms are then developed for the estimation and computation of all the above-mentioned performance indices. In order to guarantee the required estimation confidential, determination of the minimum number of the samples to be generated using the RA method is included in the estimation algorithms.

The last level of the framework is established in Chap. 18, in which a number of RA-based design schemes and algorithms for fault detection systems are proposed. While the first two levels are functionality oriented, this level is application oriented and thus open for integrating further design algorithms in future. In the first section, two randomised algorithms are proposed for threshold settings. They can be used for the threshold setting purpose for any type of systems. In fact, they have been applied to dealing with threshold setting tasks in a number of chapters in this book. In the second section, a RA-based design of observer-based fault detection systems is developed. This design scheme is similar to the design method described in Chap. 6, but can be efficiently applied to dealing with uncertainties in the system under consideration. At the end of this section, multiple monitoring indices based fault detection is under consideration. The use of multiple features expressed in terms of multiple monitoring indices is a common practice in machine learning aided fault diagnosis and could considerably improve fault detectability in comparison with a single monitoring index. For this purpose, we propose a RA-based design scheme whose core is the optimal selection of a threshold hyperplane. Applying the well-known support vector machine (SVM) technique, a RA-based solution is derived, which maximises the FDR and guarantees the pre-defined upper-bound of FAR. It is worth remarking that this is the first study on the use of multiple monitoring indices in the model-based fault detection framework in combination with SVM as the tool for the problem solution.

1.3.6 Part VI: An Integrated Framework of Control and Diagnosis, And fault-Tolerant Control Schemes

This part is composed of four chapters and focuses on analysis of control and diagnosis performance from an integrated perspective, performance and performance degradation monitoring as well as fault-tolerant control in the context of performance degradation recovery.

Chap. 19 consists of several topics related to the analysis and performance assessment of feedback control systems. In the center of our study stand, however, the observer-based residual signal and an observer-based input-output model. In this model, the residual signal plays an essential role and enables us to handle uncertainties as accessible system variables. Based on this model, new concepts related to performance assessment are introduced, and some standard control problems can be alternatively addressed in the context of performance degradation recovery. The first work in this regard is the introduction of the concept loop performance degradation (LPD) and its use in control performance monitoring. LPD is an extension and generalisation of the well-known concept loop transfer recovery (LTR) and deals with recovering control performance degradation caused, besides by the use of a state estimate (as handled in the LTR framework), by uncertainties and faults in a feedback control loop. On this basis, two assessment schemes are proposed for monitoring control performance degradation. Related to the work in Chap. 9, the role of the SIR of a feedback controller in recovering system control and detection performance is studied. As system performances are (i) stability margin, (ii) fault detectability indicator, and (iii) LPD under consideration. It is demonstrated that the optimisation of these three performance indices can be achieved uniformly by minimising the $\mathcal{H}_\infty$-norm of the SIR of the feedback controller. It is revealed that minimising the SIR is equivalent to the minimisation of the transfer function from the uncertainties or/and faults (expressed in terms of the residual signal) to the estimates for the state feedback controller. This is the unified perspective of these three system performance and is also called information and estimation perspective of control and detection.

Chap. 20 serves for four purposes concerning with performance monitoring and fault-tolerant control. Besides (i) reviewing the standard linear quadratic Gaussian (LQG) and linear quadratic regulator (LQR) (or $\mathcal{H}_2$) control problems and providing the alternative solutions based on the observer-based input-output model introduced in Chap. 19, (ii) formulating and solving performance degradation monitoring and recovering problems for feedback control loops with a linear quadratic (LQ) controller, (iii) formulating LQ optimal observer design problem, studying its solution and some relevant issues, and (iv) formulating and solving LQ observer performance degradation monitoring and recovering problems are the further objectives. This work builds the fundament for our investigation on fault-tolerant control, performance degradation recovery, and online observer optimisation in the subsequent chapters. To be specific, LQG and LQR are first handled based on the observer-based input-output model, which yields the same solutions known in the literature. It is followed by the study on the so-called LQ optimal observer design. The motivation of this work is the separation principle and the known result that the LQR controller ($\mathcal{H}_2$ controller) is composed of an LQ state feedback controller and an $\mathcal{H}_2$ observer. The latter is a special case of the LQ observer defined and addressed in our work. In fact, from the viewpoint of the cost function, LQ optimal observer is similar with the LS observer studied in Chap. 8. On the other hand, our major focus in this chapter is on (i) the role and interpretation of the co-state vector, (ii) the (smoothing) estimates of the state and unknown input vectors, which is important for the subsequent work on online optimisation of the observer, and (iii) the dual form and relations between the LQ optimal controller and observer. In the following two sections, performance degradation monitoring and recovering issues for control and estimation systems are investigated, respectively. For the purpose of detecting control performance degradation, two algorithms are proposed. In the second algorithm, the well-known Bellman equation is used as the performance residual model for performance degradation prediction. In the reinforcement learning framework, the performance residual is called temporal difference (TD). Our work on the control performance degradation recovery is inspired by the so-called Q-learning method known in the reinforcement learning technique and widely applied in online and model-free optimisation of LQ controllers. To recover the control performance, a scheme for updating state feedback gain matrix is proposed, in which the update is triggered by the prediction of performance degradation. The last section of this chapter addresses the monitoring and performance recovering issues of observers which are applied both for the control and fault detection purposes. Thanks to the duality of the LQ controller and observer, the detection issue of observer performance degradation can be then formulated and solved analogue to detecting LQ control performance degradation. The core of the observer performance degradation recovery is an estimation problem. Using the process data and the smoothing estimates of the state and unknown input variables, the variations in the system parameters are estimated. To this end, an algorithm is developed.

Chap. 21 begins with a discussion on the component oriented and performance-based fault-tolerant control strategies and underlines the needs for the latter. In this context, a schematic framework of performance-based fault-tolerant control is sketched. While the study in the previous chapter concentrates on updating the state feedback gain and observer in the control loop towards monitoring and recovering LQ control performance, the performance-based fault-tolerant control schemes presented in this chapter deal with tuning the parameter system of the Youla parameterised stabilisation controller. In the first scheme, the system performance under supervision is the stability margin. Corresponding to it, the so-called fault-tolerant margin is introduced as an indicator for the performance degradation. In this regard, an algorithm to detect performance degradation and two algorithms for performance recovery are developed. The second scheme addresses the recovery of loop performance degradation defined in Chap. 19. Based on the so-called dual form of the Youla parameterisation (of stabilisation controllers), a relation between the residual vector and the parameterised model uncertainties (can also be faults) is built. This allows us to formulate the performance degradation problem as a (robust) control problem with the parameter system of the Youla parameterised controller as a dynamic feedback controller and the uncertainties (faults) as the (unknown) plant. As a result, the performance monitoring and recovering algorithms developed in Chap. 20 can be applied for our purpose of recovering loop performance degradation.

The objective of Chap. 22 is to study fault-tolerant control issues in the data-driven fashion. Considering that online identification of the plant model during closed-loop operations is often a part of a fault-tolerant scheme, we first address the online identification issue of SIR and SKR in the closed-loop configuration. This work yields the state space models of the data-driven SIR and SKR. The most convincing argument for applying these models for performing online monitoring and control tasks is that all state variables in the models are accessible. This enables the realisation of performance monitoring and fault-tolerant control in the data-driven fashion and allows us to apply the existing algorithms presented in Chaps. 20–21 to realise performance monitoring and performance degradation recovering. In the last section of this chapter, an algorithm is developed to achieve performance degradation recovery by means of a reduced controller. In summary, our work in this chapter enables an optimal performance degradation recovery by means of a dynamic output controller with flexible structure.

1.4 Notes and References

In the last three decades, a great number of monographs on process monitoring, diagnosis and fault-tolerant control have been published. Among them, for instance, [1–10] are dedicated to the model-based fault diagnosis techniques, [11–14] are focused on the data-driven process monitoring and diagnosis methods, and [8, 15, 16] deal with fault-tolerant control issues. Concerning the surveys on these topics, we recommend the reader the following representative survey papers, [17–23] on the model-based fault diagnosis techniques, [24–31] on the data-driven process monitoring and diagnosis, and [22, 32, 33] on fault-tolerant control methods.

Although it is dedicated to the application of machine learning technique to machine fault diagnosis, the survey paper by Lei et al. [34] provides excellent insights into the application of machine learning methods to fault diagnosis and gives an elaborate summary of the basic ideas, concepts and schemes in this research domain. We strongly recommend it to the reader. It is worth mentioning that transfer learning technique [35] is viewed in this survey as the key technology of the next generation. A successful case study of applying transfer learning technique to fault diagnosis is reported in [36].

Due to its increasing importance in industrial applications, diagnosis of intermittent and incipient faults is receiving considerable research attention. The reader is referred to the review papers in [37–39] for a comprehensive description of the state of the art of the related techniques. In this book, no explicit study will be devoted to this topic, except that applicability and efficiency of some detection and performance degradation recovery methods with respect to incipient faults are discussed.

The Cover's theorem mentioned in Example 1.6 was published by Cover in [40]. Schölkopf et al. have made initial contributions to the KPCA technique [41]. In [42], an early application of the KPCA technique to fault detection has been reported. The academic example considered in Example 1.6 is adopted from [41, 42].

Fault diagnosis and fault-tolerant control is an interdisciplinary field. For a success research in this field, good knowledge of MVA and statistics, linear algebra, linear system theory and robust control theory is necessary. Throughout this book, known and well-developed methods and algorithms from these thematic areas will serve as the major tools. Among the great number of available books on these topics, we would like to recommend the following representative ones: [1, 43] on statistical methods and MVA, [44, 45] on linear algebra and matrix theory, [46, 47] on linear system theory, [48] on filtering theory, and [49] on robust control theory.

References

1. M. Basseville and I. Nikiforov, *Detection of Abrupt Changes -Theory and Application*. New Jersey: Prentice-Hall, 1993.
2. J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York Basel Hong Kong: Marcel Dekker, 1998.

3. R. Mangoubi, *Robust Estimation and Failure Detection*. London: Springer, 1998.
4. J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Boston: Kluwer Academic Publishers, 1999.
5. R. J. Patton, P. M. Frank, and R. N. C. (Eds.), *Issues of Fault Diagnosis for Dynamic Systems*. London: Springer, 2000.
6. F. Gustafsson, *Adaptive Filtering and Change Detection*. Chichester: John Wiley and Sons, LTD, 2000.
7. S. Simani, S. Fantuzzi, and R. J. Patton, *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. London: Springer-Verlag, 2003.
8. M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control, 2nd Edition*. Berlin Heidelberg: Springer, 2006.
9. R. Isermann, *Fault Diagnosis Systems*. Berlin Heidelberg: Springer-Verlag, 2006.
10. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.
11. E. L. Russell, L. Chiang, and R. D. Braatz, *Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. London: Springer-Verlag, 2000.
12. L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London: Springer, 2001.
13. Z. Q. Ge and Z. H. Song, *Multivariate statistic process control*. London: Springer-Verlag, 2013.
14. S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems*. London: Springer-Verlag, 2014.
15. M. Mahmoud, J. Jiang, and Y. Zhang, *Active Fault Tolerant Control Systems*. London: Springer, 2003.
16. H. Noura, D. Theilliol, J. Ponsart, and A. Chamseddine, *Faul-Tolerant Control Systems: Design and Practical Applications*. New York, NY, USA: Springer, 2009.
17. P. M. Frank, “Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey,” *Automatica*, vol. 26, pp. 459–474, 1990.
18. P. M. Frank and X. Ding, “Survey of robust residual generation and evaluation methods in observer-based fault detection systems,” *Journal of Process Control*, vol. 7(6), pp. 403–424, 1997.
19. V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri, “A review of process fault detection and diagnosis part I: Quantitative model-based methods,” *Computers and Chemical Engineering*, vol. 27, pp. 293–311, 2003.
20. P. Zhang and S. X. Ding, “On fault detection in linear discrete-time, periodic, and sampled-data systems (survey),” *Journal of Control Science and Engineering*, pp. 1–18, 2008.
21. R. Mangoubi, M. Desai, A. Edelmayer, and P. Sammak, “Robust detection and estimation in dynamic systems and statistical signal processing: Intersection, parallel paths and applications,” *European Journal of Control*, vol. 15, pp. 348–369, 2009.
22. I. Hwang, S. Kim, Y. Kim, and C. Seah, “A survey of fault detection, isolation, and reconfiguration methods,” *IEEE Trans. Contr. Syst. Tech.*, vol. 18, pp. 636–653, 2010.
23. Z. W. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques, part I: Fault diagnosis with model-based and signal-based approaches,” *IEEE Trans. on Industrial Electronics*, vol. 62, pp. 3757–3767, 2015.
24. V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin, “A review of process fault detection and diagnosis part III: Process history based methods,” *Computers and Chemical Engineering*, vol. 27, pp. 327–346, 2003.
25. S. J. Qin, “Statistical process monitoring: Basics and beyond,” *Journal of Chemometrics*, vol. 17, pp. 480–502, 2003.
26. S. J. Qin, “Survey on data-driven industrial process monitoring and diagnosis,” *Annual Reviews in Control*, vol. 36, pp. 220–234, 2012.
27. Z. Q. Ge, Z. H. Song, and F. R. Gao, “Review of recent research on data-based process monitoring,” *Industrial and Engineering Chemistry Research*, vol. 52, pp. 1446–1455, 2013.
28. S. X. Ding, “Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results,” *Journal of Process Control*, vol. Vol. 24, pp. 431–449, 2014.

29. S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. on Industrial Electronics*, vol. 61, pp. 6418–6428, 2014.
30. Z. W. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques, part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. on Industrial Electronics*, vol. 62, pp. 3768–3774, 2015.
31. D. Hoang and H. Kang, "A survey on deep learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327–335, 2019.
32. Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Review in Control*, vol. 32, pp. 229–252, 2008.
33. S. Yin, B. Xiao, S. X. Ding, and D. Zhou, "A review on recent development of spacecraft attitude fault-tolerant control system," *IEEE Trans. on Industrial Electronics*, vol. 63, pp. 3311–3320, 2016.
34. Y. Lei, Y. Bin, X. Jiang, F. Jia, N. Li, and A. Nandi, "Applications of machine learning to machine fault diagnosis: a review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, pp. 1–39, 2020.
35. S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, 2010.
36. B. Yang, Y. Lei, F. Jia, and S. Xiang, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," *Mech. Syst. Signal Process.*, vol. 122, pp. 692–706, 2019.
37. J. Li, D. H. Zhou, X. H. Si, M. Y. Chen, and C. H. Xu, "Review of incipient fault diagnosis methods," *Control Theory and Applications*, vol. 29, pp. 1517–1529, 2012.
38. C. Wen, F. Lv, Z. Bao, and M. Liu, "A review of data driven-based incipient fault diagnosis," *ACTA Automatica SINICA*, vol. 42, pp. 1285 – 1299, 2016.
39. D. Zhou, Y. Zhao, Z. Wang, X. He, and M. Gao, "Review on diagnosis techniques for intermittent faults in dynamic systems," *IEEE Trans. on Indus. Electronics*, vol. 67, pp. 2337 – 2347, 2020.
40. T. M. Cover, "Geometric and statistical properties of systems of linear inequalities with applications to pattern recognition," *IEEE Trans. on Electronic Computers*, vol. EC-14, pp. 326 – 334, 1965.
41. B. Schoelkopf, A. Smola, and K.-R. Mueller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, 1998.
42. J.-M. Lee, C. Yoo, S. Choi, P. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, pp. 223–234, 2004.
43. W. K. Härdle and L. Simar, *Applied Multivariate Statistical Analysis, Third Edition*. Berlin Heidelberg: Springer, 2012.
44. F. R. Gantmacher, *The Theory of Matrices*. USA: Chelsea Publishing Company, 1959.
45. G. Strang, *Introduction to Linear Algebra, 4th Edition*. Wellesley, MA: Wellesley-Cambridge Press, 2009.
46. C. T. Chen, *Linear System Theory and Design*. Holt Rinehart, Winston, 1984.
47. T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
48. T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. New Jersey: Prentice Hall, 1999.
49. K. Zhou, *Essential of Robust Control*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

Chapter 2
Basic Requirements on Fault Detection and Estimation

In this chapter, we first introduce the essential configuration of a fault detection and estimation system, and describe the fundamental requirements on such a fault detection and estimation system. On this basis, elementary fault detection and estimation problems are formulated.

2.1 Fault Detection and Estimation Paradigm

A fault detection (FD) and fault estimation (FE) system is driven by the measurements of the process under consideration. These measurements are typically sensor signals. If the process operation is regulated by some control signals, they can also be used as measurements for FD and FE purpose. In most cases, we denote the sensor signals by vector y and process control signals by u. Suppose that the fault to be detected can be modelled by a signal vector which is denoted by f and satisfies

$$\begin{cases} \text{fault-free: } f = 0, \\ \text{faulty: } f \neq 0. \end{cases}$$

The first step to a successful fault detection is to build a mapping from the measurement space, (y, u) , to an image space of the fault, f, which is then applied as a fault detector. This procedure can be schematically described by

$$J = \mathcal{J}(y, u) = \mathcal{D}(f).$$

There are numerous terms for the function J. In the field of machine learning, it is called feature, while in statistic multi-variate analysis (MVA) and model-based fault diagnosis it is often called test statistic and evaluation function, respectively.

In the model-based FD and FE framework, in particular when the process under consideration is driven by control input u, $\mathcal{D}(f)$ is often built in two steps: (i) mapping (y, u) to the so-called residual subspace with residual vector r, which is

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_2

also an image space of f, and (ii) building J as a function of r. Mathematically, these two steps can be described by

$$\text{(i) } r = \mathcal{K}(y, u) = \mathcal{Q}(f), \text{ (ii) } J = \mathcal{J}(r) = \mathcal{J}(\mathcal{Q}(f)) = \mathcal{D}(f).$$

The real challenge in dealing with fault diagnosis consists in the existence of process uncertainties, which can arise in different forms. In order to simplify our problem description, we first restrict our attention to those uncertainties which are expressed in form of additive unknown inputs including noises and (deterministic) disturbances. They corrupt the process measurements and affect the fault indicator. In order to distinguish the influences of the unknown inputs and the fault f, two basic steps are widely adopted: (i) design the function J to enhance the influence of f on the fault detector and simultaneously to degrade the influence of the unknown inputs, and (ii) introduce a threshold J_{th} and, based on $\{J, J_{th}\}$, run the (simple) detection logic,

$$\begin{cases} J - J_{th} \leq 0 \Longrightarrow \text{fault-free}, \\ J - J_{th} > 0 \Longrightarrow \text{faulty} \Longrightarrow \text{alarm}, \end{cases} \tag{2.1}$$

online for a fault detection.

Once a fault is detected, a fault estimator can be activated, when needed, which is driven either by y or (y, u), and can be schematically described by

$$\hat{f} = \mathcal{I}(y) \text{ or } \hat{f} = \mathcal{I}(y, u).$$

In summary, the major design tasks of a fault detector and a fault estimator consist of

- construction of $\mathcal{J}(y, u)$ or $\mathcal{K}(y, u)$ and $\mathcal{J}(r)$,
- setting of J_{th}, and
- construction of $\mathcal{I}(y)$ or $\mathcal{I}(y, u)$.

2.2 Fault Detection and Estimation in the Probabilistic Framework

In this section, we briefly introduce the major indicators for assessing fault detection performance in a probabilistic framework. It is followed by the formulation of basic optimal fault detection and estimation problems.

2.2.1 Fault Detection Performance Assessment

For a process with random uncertainties, false alarm rate (FAR), missed detection rate (MDR) or fault detection rate (FDR) are widely adopted for assessing FD performance. They are defined as follows.

Definition 2.1 *Given the test statistic J, threshold J_{th} and detection logic (2.1), FAR is defined as the probability*

$$FAR = \Pr\left(J > J_{th} \,|f = 0\right). \tag{2.2}$$

Definition 2.2 *Given the test statistic J, threshold J_{th} and detection logic (2.1), MDR is defined as the probability*

$$MDR = \Pr\left(J \leq J_{th} \,|f \neq 0\right). \tag{2.3}$$

Definition 2.3 *Given the test statistic J, threshold J_{th} and detection logic (2.1), FDR is defined as the probability*

$$FDR = \Pr\left(J > J_{th} \,|f \neq 0\right). \tag{2.4}$$

It is clear that it holds

$$FDR = 1 - MDR.$$

Hence, both of them, MDR and FDR, can be used for assessing fault detectability.

It should be noticed that the computation of FAR and MDR/FDR requires knowledge of the distribution of the test statistic J *a priori*, both in the faulty and fault-free cases. Also, FDR/MDR is a function of f. In Chaps. 16–17, we shall discuss about their computation in the probabilistic framework and introduce the so-called randomised algorithms technique to deal with relevant computation issues.

2.2.2 Optimal Fault Detection and Estimation Problems

An FD problem is defined by finding (i) a test statistic J, and (ii) a corresponding threshold J_{th}. Considering that the test statistic J is a function of the fault to be detected and random uncertainties, reducing FAR may lead to increasing MDR, and vise versa. Thus, an optimal FD solution is, in fact, an optimal trade-off between FAR and MDR/FDR. This motivates us to formulate two alternative optimisation problems.

Definition 2.4 *FD with maximum fault detectability: A solution to this FD problem is said to be optimal, when, for a given acceptable* $FAR\ \alpha$,

$$\{J, J_{th}\} = \arg \min_{J, J_{th}, FAR \leq \alpha} MDR. \tag{2.5}$$

Definition 2.5 *FD with minimum false alarm rate: A solution to this FD problem is said to be optimal, when, for a given acceptable* $MDR\ \beta$,

$$\{J, J_{th}\} = \arg \min_{J, J_{th}, MDR \leq \beta} FAR. \tag{2.6}$$

Major attention in theoretical studies and applications has been devoted to the problem of FD with maximum fault detectability. In fact, FD with minimum false alarm rate is its dual form. In our subsequent work, we shall focus on FD with maximum fault detectability and handle FD with minimum false alarm rate as a dual form of the first FD problem.

Concerning FE, we call $\hat{f}$ an optimal estimation of f if

- it is unbiased, that is

$$\mathcal{E}\left(\hat{f}\right) = f, \tag{2.7}$$

- and LMS (least mean squares) defined by

$$\hat{f} = \arg \min_{\bar{f}, \mathcal{E}(\bar{f}) = f} \mathcal{E}\left(f - \bar{f}\right)^T \left(f - \bar{f}\right). \tag{2.8}$$

2.3 Fault Detection and Estimation in Deterministic Processes

In the model-based framework, FD and FE in processes with deterministic unknown inputs have been extensively investigated. On the other hand, there do not exist common criteria for the performance assessment. The design problems are often formulated in form of certain FD or FE system specifications like sensitivity or robustness. This section is dedicated to (i) introduction of some concepts for the performance assessment and, based on them, (ii) formulation of basic optimal FD and FE problems.

2.3.1 Performance Assessment

For FD in processes with deterministic unknown inputs, FAR,MDR/FDR are not applicable for assessing FD performance. This calls for alternative performance indicators.

Let y be the process (output) measurement vector with

$$y = \mathcal{M}_d(d) + \mathcal{M}_f(f) = y_d + y_f, \tag{2.9}$$

where $\mathcal{M}_d : \mathcal{D}_d \to \mathcal{D}_y, \mathcal{M}_f : \mathcal{D}_f \to \mathcal{D}_y$ represent the mappings of disturbance vector $d \in \mathcal{D}_d$ and fault vector $f \in \mathcal{D}_f$ to the measurement vector $y \in \mathcal{D}_y$, respectively. $\mathcal{D}_d, \mathcal{D}_f$ and $\mathcal{D}_y$ are general notations for the domains of the disturbances, faults and measurements. For instance, for static processes, $\mathcal{M}_d, \mathcal{M}_f$ would be constant matrices and $\mathcal{D}_d = \mathcal{R}^{k_d}, \mathcal{D}_f = \mathcal{R}^{k_f}$ and $\mathcal{D}_y = \mathcal{R}^m$, and for LTI processes, they could be transfer functions or the system state space representations and $\mathcal{D}_d = \mathcal{H}_2^{k_d}, \mathcal{D}_f = \mathcal{H}_2^{k_f}$ and $\mathcal{D}_y = \mathcal{H}_2^m$ if the process under consideration is stable.

Remark 2.1 *A more general form of the measurement model can be described as*

$$y = \mathcal{M}(d, f)$$

with

$$y = \begin{cases} \mathcal{M}(d, 0) = \mathcal{M}_d(d) = y_d, \text{ fault-free,} \\ \mathcal{M}(0, f) = \mathcal{M}_f(f) = y_f, \text{ faulty, uncertain-free.} \end{cases}$$

On the other hand, in most of our studies, linear systems are under consideration, which are well represented by the model (2.9). In addition, in our subsequent work the fault-free and faulty operations are often separately addressed. In this regard, we could also consider the following theoretical model

$$y = \begin{cases} y_d = \mathcal{M}_d(d), \text{ fault-free,} \\ y_f = \mathcal{M}_f(f), \text{ faulty.} \end{cases}$$

We denote the images of $\mathcal{M}_d, \mathcal{M}_f$ by

$$\mathcal{I}_{\mathcal{M}_d} = \{y_d \,|y_d = \mathcal{M}_d(d), \forall d \in \mathcal{D}_d\},$$
$$\mathcal{I}_{\mathcal{M}_f} = \left\{y_f \,\middle|y_f = \mathcal{M}_f(f), \forall f \in \mathcal{D}_f\right\},$$

and assume, at first,

$$\mathcal{I}_{\mathcal{M}_d} = \mathcal{I}_{\mathcal{M}_f}. \tag{2.10}$$

That means

$$\forall y_d \in \mathcal{I}_{\mathcal{M}_d}, \exists f \text{ s.t. } y_f = \mathcal{M}_f(f) = y_d,$$

and vice versa. This implies as well, the images of $\mathcal{M}_d$, $\mathcal{M}_f$ have the same dimension equal to the one of y,

$$\dim(y_d) = \dim(y_f) = \dim(y). \tag{2.11}$$

Note that the relation (2.10)–(2.11) can be understood as a total overlapping of the influences of d and f in the measurement space.

Inspired by the concept of rejection and critical regions used in hypothesis tests and the (linear) separability of two sets adopted in the classification technique, we will first define various sets related to the influences of d and f.

We define the image of the disturbance vector as

$$\mathcal{I}_d = \{y_d \, | y_d = \mathcal{M}_d(d), \forall d \in \mathcal{D}_d, \|d\|_N \leq \delta_d\}, \tag{2.12}$$

where $\|\cdot\|_N$ is the placeholder for a certain norm of d with δ_d as its boundedness. Note that due to the boundedness of d, $\mathcal{I}_d$ is, in general, a subspace of the image of $\mathcal{M}_d$,

$$\mathcal{I}_d \subset \mathcal{I}_{\mathcal{M}_d}.$$

From the classification point of view, it is clear that a fault f is not separable from the disturbance if its influence on y, $y_f = \mathcal{M}_f(f)$, belongs to $\mathcal{I}_d$. In this context, we introduce the definition of the set of undetectable faults.

Definition 2.6 *The fault domain defined by*

$$\mathcal{D}_{f,undetc} = \left\{f \left| f \in \mathcal{D}_f, y_f = \mathcal{M}_f(f) \in \mathcal{I}_d \right.\right\} \tag{2.13}$$

is called the set of undetectable faults.

It is clear that an optimal fault detection is achieved in the sense of maximal fault detection rate, when all faults that do not belong to the set of undetectable faults can be detected.

Definition 2.7 *A solution $\{J, J_{th}\}$ to the FD problem with the presence of disturbances d is called optimal in the sense of maximising the fault detectability, when*

$$\forall y \in \mathcal{I}_d, f = 0, J(y) - J_{th} \leq 0, \tag{2.14}$$

$$\forall f \notin \mathcal{D}_{f,undetc}, d = 0, J(y) - J_{th} > 0. \tag{2.15}$$

It should be remarked that such an optimal solution leads to the maximum fault detectability and, simultaneously, no false alarm. This type of FD problems has been mostly investigated. In comparison, less attention has been paid to its dual form, which is briefly formulated below.

Suppose that we are only interested in detecting those faults whose norm, $\|f\|_N$, is larger than a pre-defined level. In other words, if $\|f\|_N$ is lower than the pre-defined level, an alarm should not be triggered.

Definition 2.8 *Given constant $\beta > 0$, the domain*

$$\mathcal{D}_{f,\beta} = \left\{f \left| f \in \mathcal{D}_f, \|f\|_N \leq \beta \right.\right\} \tag{2.16}$$

is called the set of faults of no interest. β is called the margin of detectable faults.

To simplify our study, we assume that

$$\forall f \in \mathcal{D}_f,\ f \neq 0,\ y_f = \mathcal{M}_f(f) \neq 0,$$

which means, the dimension of the kernel space of $\mathcal{M}_f$ is zero and equivalently

$$\dim(f) = \dim\left(y_f\right) \leq \dim(y). \tag{2.17}$$

This assumption loses no practical applicability and can be interpreted as no need to detect those faults which cause no change in the process output.

In order to reduce false alarms, d should not trigger an alarm if its response y_d belongs to $\mathcal{I}_{f,\beta}$,

$$\mathcal{I}_{f,\beta} = \left\{ y_f \left| y_f = \mathcal{M}_f(f),\ f \in \mathcal{D}_{f,\beta} \right. \right\}.$$

That is, there exists $f \in \mathcal{D}_{f,\beta}$ so that

$$y_f = \mathcal{M}_f(f) = y_d.$$

Hence, for our purpose, we define the following disturbance image set

$$\mathcal{I}_{d,\beta} = \left\{ y_d \left| y_d = \mathcal{M}_d(d) \in \mathcal{I}_{f,\beta},\ d \in \mathcal{D}_d \right. \right\}. \tag{2.18}$$

For a reliable FD, it is of interest that those disturbances whose responses belong to $\mathcal{I}_{d,\beta}$ should not trigger alarms. In this context, we introduce the dual form of the FD problem defined in Definition 2.7.

Definition 2.9 *A solution* $\{J, J_{th}\}$ *to the FD problem with the presence of disturbances* d *is called optimal in the sense of minimising the number of false alarms, when*

$$\forall f \notin \mathcal{D}_{f,\beta},\ d = 0,\ J(y) - J_{th} > 0, \tag{2.19}$$

$$\forall y \in \mathcal{I}_{d,\beta},\ f = 0,\ J(y) - J_{th} \leq 0. \tag{2.20}$$

Condition (2.19) gives a necessary condition for detecting a fault "of interest" without taking into account the influence of disturbances, which can trigger false alarms. In order to reduce the number of false alarms as much as possible, condition (2.20) is introduced to avoid false alarms which may be, otherwise, caused by those disturbances whose responses belong to $\mathcal{I}_{d,\beta}$.

Once a fault is detected, a fault estimation can be realised by finding a generalised inverse of $\mathcal{M}_f$. In our work, an optimal FE for f is understood as a least squares (LS) estimation and will be addressed for various types of processes.

2.3.2 Characterisation of Optimal Solutions

In this sub-section, we give some results, which characterise the solutions to the formulated FD problems and will become useful in our subsequent study on the design of FD systems for different types of process models.

Theorem 2.1 *Let $\mathcal{M}_d^-$ be an operator satisfying the following conditions:*

(i) $\mathcal{M}_d^-$ *is (left) invertible,*
(ii) $\forall d \in \mathcal{D}_d$

$$\left\| \mathcal{M}_d^- \circ \mathcal{M}_d(d) \right\|_N \leq \|d\|_N \,, \tag{2.21}$$

(iii) for f satisfying

$$\left\| \mathcal{M}_d^- \circ \mathcal{M}_f(f) \right\|_N \leq \delta_d,$$

there exists d, $\|d\|_N \leq \delta_d$, so that

$$\mathcal{M}_d^- \circ \mathcal{M}_f(f) = \mathcal{M}_d^- \circ \mathcal{M}_d(d). \tag{2.22}$$

Then, $\{J, J_{th}\}$ given by

$$J = \|r\|_N^2 \,, r = \mathcal{M}_d^- y, \, J_{th} = \delta_d^2 \tag{2.23}$$

solves the optimal FD problem defined in Definition 2.7.

Before proving the theorem, we would like to explain and understand the above three conditions. Note that under condition (i), (2.11) holds true. This ensures that there is no change in the dimension of image space of the faults and thus no loss of information about the faults to be detected. Conditions (ii) and (iii) can be understood as the fact that $\mathcal{M}_d^-$ should be a generalised inverse of $\mathcal{M}_d$.

Proof It is evident that for $f = 0$

$$J - J_{th} = \left\| \mathcal{M}_d^- \circ \mathcal{M}_d(d) \right\|_N^2 - \delta_d^2 \leq 0.$$

Thus, (2.14) holds. We now prove condition (2.15) by contradiction. Assume that $f \notin \mathcal{D}_{f,undetc}$ but $J \leq J_{th}$ for $d = 0$. It turns out

$$J = \left\| \mathcal{M}_d^- \circ \mathcal{M}_f(f) \right\|_N^2 \leq \delta_d^2.$$

Following condition (iii), $\exists d$, $\|d\|_N \leq \delta_d$, s.t.

$$\mathcal{M}_d^- \circ \mathcal{M}_f(f) = \mathcal{M}_d^- \circ \mathcal{M}_d(d),$$

and furthermore

$$y_d = \mathcal{M}_d(d) \in \mathcal{I}_d.$$

Considering further condition (i), we have

$$\mathcal{M}_f(f) = \mathcal{M}_d(d) = y_d,$$

which means, in turn,

$$y_f = \mathcal{M}_f(f) \in \mathcal{I}_d \Longrightarrow f \in \mathcal{I}_{f,undetc.}$$

Thus, by contradiction, it is proved that (2.15) holds.

The three conditions given in Theorem 2.1 characterise the major features of the optimal solutions and can be viewed as a guideline for designing FD systems.

At this point, we would like to call reader's attention to the fact that $\mathcal{M}_d^-$ is a generalised inverse of $\mathcal{M}_d$, which will also be demonstrated in our subsequent study on different types of systems. This fact allows us to interpret $\mathcal{M}_d^- y_d$ as an LS estimation of d. In other words, the optimal FD problem can also be viewed as an optimal estimation problem for the disturbance d. Indeed, this view of the FD problem is not surprising, since all available information about d is its norm-boundedness. It is reasonable to apply the norm of the LS estimate of d to build the evaluation function and to compare it with the known norm-boundedness δ_d^2 (as threshold).

Example 2.1 *In order to illustrate Theorem 2.1, we consider a simple static system described by*

$$\begin{aligned} y &= \mathcal{M}_d(d) + \mathcal{M}_f(f) = M_d d + M_f f \in \mathcal{R}^m, \\ rank\,(M_d) &= rank\left(M_f\right) = m, d \in \mathcal{R}^{k_d}, f \in \mathcal{R}^{k_f}, \end{aligned}$$

where $M_d \in \mathcal{R}^{m \times k_d}$, $M_f \in \mathcal{R}^{m \times k_f}$ are constant matrices. Let

$$\mathcal{M}_d^- = M_d^- = M_d^T \left(M_d M_d^T\right)^{-1}.$$

That is, $\mathcal{M}_d^-$ is the generalised inverse of M_d. It is straightforward that M_d^- is left invertible,

$$M_d M_d^- = I,$$

and

$$\left\| M_d^T \left(M_d M_d^T\right)^{-1} M_d d \right\| \leq \|d\|$$

so that condition (2.21) is satisfied, where the Euclidean norm is adopted as $\|\cdot\|_N$, that is

$$\|\cdot\|_N = \|\cdot\|.$$

Moreover, if

$$\left\| M_d^T \left(M_d M_d^T\right)^{-1} M_f f \right\|^2 \leq \delta_d^2 = J_{th},$$

we have, for $d = M_d^T \left(M_d M_d^T\right)^{-1} M_f f$,

$$\|d\| = \left\| M_d^T \left(M_d M_d^T\right)^{-1} M_f f \right\| \leq \delta_d$$

so that

$$\begin{aligned} \mathcal{M}_d^- \mathcal{M}_f(f) &= M_d^T \left(M_d M_d^T\right)^{-1} M_f f \\ &= M_d^T \left(M_d M_d^T\right)^{-1} M_d M_d^T \left(M_d M_d^T\right)^{-1} M_f f \\ &= \mathcal{M}_d^- \mathcal{M}_d(d). \end{aligned}$$

Thus, condition (2.22) holds.

As a solution to the FD problem defined in Definition 2.9, we give the following corollary which is a dual result of Theorem 2.1.

Corollary 2.1 *Given the margin of detectable faults* β. *Let* $\mathcal{M}_f^-$ *satisfy*

$$\forall f \in \mathcal{D}_f, y_f = \mathcal{M}_f^- \circ \mathcal{M}_f(f), \left\| y_f \right\|_N = \|f\|_N. \tag{2.24}$$

Then, $\{J, J_{th}\}$ *given by*

$$J = \|r\|_N^2, r = \mathcal{M}_f^- y, J_{th} = \beta^2 \tag{2.25}$$

solves the optimal FD problem defined in Definition 2.9.

Proof Condition (2.24) guarantees that for $f \notin \mathcal{D}_{f,\beta}$, i.e. $\|f\|_N > \beta$ and $d = 0$,

$$J(y) - J_{th} = \|r\|_N^2 - \beta^2 > 0.$$

If $y_d \in \mathcal{I}_{d,\beta}$, then there exists, according to the definition of $\mathcal{I}_{d,\beta}$, $f \in \mathcal{D}_{f,\beta}$ so that

$$y_f = \mathcal{M}_f(f) = y_d,$$

which leads to

$$r = \mathcal{M}_f^- (y_d) = \mathcal{M}_f^- \mathcal{M}_f(f) \Longrightarrow \|r\|_N^2 = \|f\|_N^2 \leq \beta^2.$$

As a result, (2.20) holds.

Note that $\mathcal{M}_f^-$ is the generalised inverse of $\mathcal{M}_f$. Thus, the FD problem with respect to minimising the number of false alarms, as defined in Definition 2.9, is, in fact, an optimal (LS) estimation problem.

Remark 2.2 *We would like to point out that the result given in Corollary 2.1 is achieved based on the assumption*

$$\forall f \in \mathcal{D}_f, f \neq 0, y_f = \mathcal{M}_f(f) \neq 0$$

which leads to (2.17).

Example 2.2 *To demonstrate the duality and the result given in Corollary 2.1, we consider the same system model as given in Example 2.1 but with*

$$rank\left(M_f\right) = k_f.$$

Let

$$\mathcal{M}_f^- = M_f^- = \left(M_f^T M_f\right)^{-1} M_f^T$$

be the generalised inverse of M_f. Since

$$\mathcal{M}_f^- \mathcal{M}_f = M_f^- M_f = I,$$

it holds

$$\left\| \mathcal{M}_f^- \mathcal{M}_f(f) \right\| = \|f\|,$$

and thus condition (2.24) is satisfied. With the evaluation function and threshold defined in (2.25), it is evident that

$$\begin{aligned} &\forall f \notin \mathcal{D}_{f,\beta} = \left\{ f \left| f \in \mathcal{D}_f, \|f\| \le \beta \right. \right\}, d = 0, J > J_{th}, \\ &\forall y_d \in \mathcal{I}_{d,\beta} = \left\{ y_d \left| y_d = \mathcal{M}_d(d) \in \mathcal{I}_{f,\beta}, d \in \mathcal{D}_d \right. \right\}, f = 0, J \le J_{th}. \end{aligned}$$

That means, the two conditions in Definition 2.9 are satisfied and thus the evaluation function and threshold defined in (2.25) solve the (dual) FD optimal problem described in Definition 2.9.

2.3.3 A General Form of Problem Formulation

Recall our assumption on the identical image dimensions of $\mathcal{M}_d$ and $\mathcal{M}_f$. If the dimension of f (the number of the faults) is lower than the one of y, i.e. if the image of $\mathcal{M}_f$ only spans a subspace in the measurement space, it is reasonable to detect and estimate the faults just using the measurement in the subspace spanned by the image of $\mathcal{M}_f$. To this end, we can first find $\mathcal{M}_f^-$ satisfying

$$\forall f, \exists \hat{f} = \mathcal{M}_f^- \circ \mathcal{M}_f(f), \text{ s.t. } \left\| \hat{f} \right\|_N = \|f\|_N , \tag{2.26}$$

$$\dim\left(\hat{f}\right) = \dim(f) = \dim(\bar{y}_d), \bar{y}_d = \mathcal{M}_f^- \circ \mathcal{M}_d(d), \tag{2.27}$$

and transform the original fault detection problem into

$$\bar{y} = \mathcal{M}_f^- y = \hat{f} + \mathcal{M}_f^- \circ \mathcal{M}_d(d),$$

which is identical with model (2.9) satisfying condition (2.10). It is remarkable that in this case an FE is embedded in the FD solution.

2.4 Notes and References

The major objective of this chapter is to introduce the basic criteria for assessing a fault detection system and, associated with them, to formulate the basic optimal detection problems, which should be common for most types of technical processes. While the concepts like FAR, MDR and FDR are widely accepted both in the application and research domains as dealing with fault detection in statistic or stochastic processes, few methods have been published on the performance assessment for processes with deterministic uncertainties. Also for this reason, our major focus is on this topic.

In its core, fault detection is a process of making a decision for one of two situations: faulty or fault-free, as described in (2.1). In dealing with statistical processes, such a decision process is equivalent with statistical hypothesis testing, a well-established statistical method. Suppose "fault-free" is the null hypothesis and "faulty" the alternative hypothesis, then FAR is equivalent to the so-called "Type I Error", while MDR corresponds to the "Type II Error". For more details, the reader is referred to [2]. In fact, due to this relationship, we are able to apply the well-established theory of statistical hypothesis testing to solving some corresponding fault detection problems in the sequel.

Unfortunately, there exists no similar framework for assessing fault detection systems for processes with deterministic uncertainties, although some efforts have been reported in [1]. Sect. 2.3 is dedicated to this topic. Inspired by the concepts of rejection and critical regions adopted in hypothesis testing framework [2] and the (linear) separability of two sets adopted in the classification technique in machine learning, we have introduced the concepts like set of undetectable faults $\mathcal{D}_{f,undetc}$, image (set) of the disturbance vector $\mathcal{I}_d$, set of faults of no interest $\mathcal{D}_{f,\beta}$. Based on them, we are able to define two basic optimal fault detection problems, which are analogue to the basic fault detection problems in the probabilistic context.

It should be remarked that

- an abstract form has been adopted to describe those new concepts related to fault detection in deterministic processes so that they can be applied to various types of process models, including static and dynamic, time invariant and time varying, linear and nonlinear, as will be done in the subsequent study,
- the concepts, the fault detection problem formulations and the conceptual solutions presented in this chapter are only applicable for the so-called additive faults [1]. Different concepts and schemes will be proposed, as multiplicative or parameter faults [1] are addressed,
- often, optimal fault detection problems can be viewed and solved as an estimation problem, as pointed out in Sub-sections 2.3.2 and 2.3.3.

References

1. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.
2. E. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*. Springer, 2008.

Chapter 3
Basic Methods for Fault Detection and Estimation in Static Processes

This chapter serves for two purposes. Beside the review of basic fault detection and estimation methods, we would like to highlight and discuss about the basic ideas and concepts of fault detection and estimation, which can be, as dealing with static processes, well addressed and explained using common statistic and linear algebraic methods.

We will briefly study fault detection and estimation in processes either with noises or with deterministic disturbances, and address the relevant issues both in the model-based and data-driven fashions.

3.1 A Basic Fault Detection and Estimation Problem

We begin with a basic fault detection and estimation problem: Given

$$y = f + \varepsilon \in \mathcal{R}^m, m \geq 1, \tag{3.1}$$

where y represents the measurement vector and $\varepsilon \sim \mathcal{N}(0, \Sigma)$ the measurement noise with known covariance matrix $\Sigma > 0$, and f satisfying

$$f = 0 \Longrightarrow \text{fault-free}, \ f \neq 0 \Longrightarrow \text{faulty}, \tag{3.2}$$

denotes the fault vector, find

- an optimal solution for the fault detection problem (2.5), and
- an optimal solution for the fault estimation problem (2.7)–(2.8).

It follows from the well-known Neyman-Pearson Lemma that the use of likelihood ratio (LR) leads to the optimal solution as defined in (2.5), when f is known. For our purpose of detecting unknown (constant) fault vector f, we

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_3

- build the log-LR, which is, considering that $y \sim \mathcal{N}(f, \Sigma)$,

$$\begin{aligned} LR &= \ln \frac{L\left(f \neq 0 \,|y\right)}{L\left(f = 0 \,|y\right)} \\ &= \frac{1}{2}\left(y^T \Sigma^{-1} y - (y - f)^T \Sigma^{-1} (y - f)\right); \end{aligned}$$

- maximise $L\left(f \neq 0 \,|y\right)$ and so the LR by finding an estimation of f, which leads to

$$\hat{f} = \arg\max_{f} \frac{1}{2}\left(-\left(y - f\right)^T \Sigma^{-1} (y - f)\right) = y; \tag{3.3}$$

- build the test statistic based on the maximal LR

$$\begin{aligned} \max_{f} \frac{1}{2}\left(y^T \Sigma^{-1} y - (y - f)^T \Sigma^{-1} (y - f)\right) &= \frac{1}{2} y^T \Sigma^{-1} y \\ \Longrightarrow J = y^T \Sigma^{-1} y \sim \chi^2(m) & \qquad (3.4) \end{aligned}$$

- and finally, set the threshold

$$J_{th} = \chi^2_{\alpha}, \tag{3.5}$$

$$\Pr\left\{\chi^2(m) > \chi^2_{\alpha}\right\} = \alpha \Longleftrightarrow \Pr\left\{\chi^2(m) \leq \chi^2_{\alpha}\right\} = 1 - \alpha \tag{3.6}$$

for a given acceptable $FAR = \alpha$.

This solution is called generalised likelihood ratio (GLR) method, and the test statistic (3.4) is subject to χ^2 distribution.

Remark 3.1 *Neyman-Pearson Lemma is dedicated to performing hypothesis tests with the LR as the most powerful test at the significant level α for a given threshold. For our application, as described in the above procedure, modifications are made to match the common use in the fault diagnosis framework without loss of the applicability.*

Remark 3.2 *For the sake of simplicity, factor $\frac{1}{2}$ is omitted by defining the test statistic in (3.4).*

In fact, the optimal solution for $\{J, J_{th}\}$ given in (3.4)–(3.5) is well-known. We would like to call the reader's attention to the following interesting facts embedded in the approach to the solution:

- an (optimal) estimation of the fault, the maximal likelihood estimate (MLE) $\hat{f}$, is embedded in the decision making, as described in (3.3);
- on the other hand, a direct use of $\hat{f}^T \hat{f}$ as a test statistic leads to poor FD performance, since equation $\hat{f}^T \hat{f} = y^T y$ is not an optimal test statistic;
- however, once a fault is detected, $\hat{f} = y$ delivers an estimate for f, which is optimal in the sense of (2.7)–(2.8).

Next, we study the test statistic (3.4) aiming at understanding why this test statistic delivers an optimal solution.

Let us do an SVD on Σ,

$$\Sigma = Pdiag\left(\sigma_1^2, \cdots, \sigma_m^2\right) P^T, PP^T = I, P = \left[p_1 \cdots p_m \right]. \tag{3.7}$$

The column vectors of P, $p_1, \cdots, p_m$, span the measurement (vector) space, and each of them defines a direction of this space. Corresponding to them, $\sigma_1, \cdots, \sigma_m$ are understood as the "amplitude" (size) of the variance in each direction. Considering that larger variance means stronger uncertainty, σ_i can be interpreted as the strength of the uncertainty in the direction p_i.

Using the SVD of Σ, the test statistic J can be written into

$$J = y^T \Sigma^{-1} y = y^T Pdiag\left(\sigma_1^{-2}, \cdots, \sigma_m^{-2}\right) P^T y. \tag{3.8}$$

Note that any fault can be written as a linear combination of $p_1, \cdots, p_m$. Let

$$f = \sum_{i=1}^{m} p_i \bar{f}_i = P\bar{f}, \bar{f} = \begin{bmatrix} \bar{f}_1 \\ \vdots \\ \bar{f}_m \end{bmatrix}.$$

In the presence of fault f, the χ^2 test statistic satisfies

$$\begin{aligned} \mathcal{E}J &= \mathcal{E}\left(y^T \Sigma^{-1} y\right) = f^T \Sigma^{-1} f + \mathcal{E}\left(\varepsilon^T \Sigma^{-1} \varepsilon\right) = \sum_{i=1}^{m} \frac{\bar{f}_i^2}{\sigma_i^2} + m \\ &= \sum_{i=1}^{m} w_i \bar{f}_i^2 + m, w_i = \frac{1}{\sigma_i^2}. \end{aligned} \tag{3.9}$$

Equation (3.9) reveals that

- viewing w_i as a weighting factor, the fault will be stronger weighted in those directions, where uncertainties are weaker,
- the fault detectability can be improved in sense of enhancing FDR, when the process uncertainties can be reduced.

3.2 A General Form of Fault Detection and Estimation Problem

We now consider a general form of the above basic fault detection and estimation problem with the measurement model

$$y = E_f f + \varepsilon \in \mathcal{R}^m, \tag{3.10}$$

where f, ε are as defined previously, $E_f \in \mathcal{R}^{m \times k_f}$ is a known matrix and satisfies

$$rank\left(E_f\right) = k_f < m, \tag{3.11}$$

which means that the image subspace of the fault vector builds a k_f -dimensional subspace in the m-dimensional measurement space.

To approach an optimal fault detection in the sense of minimising MDR, we apply the GLR method and solve the detection problem by

- building the log-LR, which is, considering that $y \sim \mathcal{N}(E_f f, \Sigma)$,

$$LR = \frac{1}{2}\left(y^T \Sigma^{-1} y - \left(y - E_f f\right)^T \Sigma^{-1} (y - E_f f)\right),$$

- maximising the LR by finding an estimation of f,

$$\hat{f} = \arg\max_f \frac{1}{2}\left(-\left(y - E_f f\right)^T \Sigma^{-1} (y - E_f f)\right), \tag{3.12}$$

- building the test statistic based on the maximal LR

$$J = 2\max_f \frac{1}{2}\left(y^T \Sigma^{-1} y - \left(y - E_f f\right)^T \Sigma^{-1} (y - E_f f)\right), \tag{3.13}$$

- and finally setting the threshold

$$\Pr\left\{J > J_{th}\right\} = \alpha, \tag{3.14}$$

for a given acceptable $FAR = \alpha$.

Concretely, the solution for the maximisation problem is given by

$$\hat{f} = \arg\min_f \left(y - E_f f\right)^T \Sigma^{-1} (y - E_f f) = E_f^- y, \tag{3.15}$$

$$E_f^- = \left(E_f^T \Sigma^{-1} E_f\right)^{-1} E_f^T \Sigma^{-1}. \tag{3.16}$$

Note that $\hat{f}$ given in (3.15) is a least mean squares (LMS) estimation of f satisfying

$$\begin{aligned}\mathcal{E}\left(f - \hat{f}\right)\left(f - \hat{f}\right)^T &= \mathcal{E}\left(f - E_f^- y\right)\left(f - E_f^- y\right)^T \\ &= \mathcal{E}\left(E_f^- \varepsilon\right)\left(E_f^- \varepsilon\right)^T = \left(E_f^T \Sigma^{-1} E_f\right)^{-1}.\end{aligned}$$

It turns out

$$J = y^T \left(\Sigma^{-1} - \left(I - E_f E_f^- \right)^T \Sigma^{-1} \left(I - E_f E_f^- \right) \right) y. \tag{3.17}$$

By noting the relations

$$\Sigma^{-1} - \left(I - E_f E_f^- \right)^T \Sigma^{-1} \left(I - E_f E_f^- \right) = \Sigma^{-1} E_f \left(E_f^T \Sigma^{-1} E_f \right)^{-1} E_f^T \Sigma^{-1},$$

$$\Sigma^{-1} E_f \left(E_f^T \Sigma^{-1} E_f \right)^{-1} E_f^T \Sigma^{-1} = \left(E_f^- \right)^T E_f^T \Sigma^{-1} E_f E_f^-,$$

$$E_f^T \Sigma^{-1} E_f = \left(E_f^- \Sigma \left(E_f^- \right)^T \right)^{-1}, E_f^- y \sim \mathcal{N} \left(0, E_f^- \Sigma \left(E_f^- \right)^T \right),$$

we have

$$y^T \Sigma^{-1} E_f \left(E_f^T \Sigma^{-1} E_f \right)^{-1} E_f^T \Sigma^{-1} y = y^T \left(E_f^- \right)^T \left(E_f^- \Sigma \left(E_f^- \right)^T \right)^{-1} E_f^- y$$

$$\Longrightarrow \max_f J \sim \chi^2(k_f). \tag{3.18}$$

As a result, the threshold is finally set as

$$J_{th} = \chi_\alpha^2, \Pr \left\{ \chi^2(k_f) \leq \chi_\alpha^2 \right\} = 1 - \alpha. \tag{3.19}$$

For our purpose, we would like to present an alternative way of solving the above fault detection problem, as described in Sub-section 2.3.3. To this end, we multiply y by E_f^-, which yields

$$E_f^- y = f + E_f^- \varepsilon =: \bar{y} \in \mathcal{R}^{k_f}, E_f^- \varepsilon \sim \mathcal{N} \left(0, E_f^- \Sigma \left(E_f^- \right)^T \right). \tag{3.20}$$

The new model (3.20) is of the same form like (3.1). Thus, applying the result given in the last section leads to the test statistic

$$J = y^T \left(E_f^- \right)^T \left(E_f^- \Sigma \left(E_f^- \right)^T \right)^{-1} E_f^- y,$$

which is (3.17), and the corresponding threshold as given in (3.18).

Recall that, in regarding to the noise,

$$\Sigma \in \mathcal{R}^{m \times m}, E_f^- \Sigma \left(E_f^- \right)^T \in \mathcal{R}^{k_f \times k_f}, k_f < m.$$

That means by mapping $E_f^- : \mathcal{R}^m \to \mathcal{R}^{k_f}$ the measurement space is transformed to a lower dimensional subspace. Such a dimension reduction reduces the influence of the noise on fault detection performance without loss of information on faults, and therefore improves the fault detection performance.

It is worth remarking that the LMS estimation of f,

$$\hat{f} = E_f^- y,$$

can be used for the fault estimation purpose, once the fault is detected.

3.3 Application of Canonical Correlation Analysis to Fault Detection

In the previous sections, a basic fault detection problem and its extended form have been reviewed, in which it is assumed that the data are collected from a set of sensors modelled in form of (3.1) or (3.10). In practice, we may often have two data sets which are, for instance, from two different parts in a process or even from two different processes. Fault detection and estimation in such a process configuration are of considerable practical interests. To deal with such issues, canonical correlation analysis (CCA) can serve as a powerful tool. CCA is a well-established MVA method. In this section, we first briefly introduce the essential ideas and computations of CCA, and then discuss about their applications to fault detection and estimation.

3.3.1 *An Introduction to CCA*

CCA is a statistical method to analyse correlation relations between two random vectors. Suppose that $y \in \mathcal{R}^m, x \in \mathcal{R}^n$ are two random vectors satisfying

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathcal{E}(x) \\ \mathcal{E}(y) \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right), \Sigma_{yx} = \Sigma_{xy}^T. \tag{3.21}$$

Let J, L define some linear mappings of x and y. Roughly speaking, CCA is dedicated to finding those linear mappings that deliver the most closed correlations between x and y. As the basis for the correlation assessment, matrix

$$K = \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} \tag{3.22}$$

is taken into account. Assume that

$$rank\left(\Sigma_{xy}\right) = rank\left(K\right) = \kappa.$$

An SVD of K results in

$$K = R\Sigma V^T, \Sigma = \begin{bmatrix} diag(\sigma_1, \cdots, \sigma_\kappa) & 0 \\ 0 & 0 \end{bmatrix}, 1 \geq \sigma_1 \geq \cdots \geq \sigma_\kappa > 0, \quad (3.23)$$
$$R = \begin{bmatrix} r_1 \cdots r_n \end{bmatrix}, R^T R = I, V = \begin{bmatrix} v_1 \cdots v_m \end{bmatrix}, V^T V = I.$$

Let

$$J = \Sigma_x^{-1/2} R, J = \begin{bmatrix} J_1 \cdots J_n \end{bmatrix}, L = \Sigma_y^{-1/2} V, L = \begin{bmatrix} L_1 \cdots L_m \end{bmatrix}. \quad (3.24)$$

It is evident that

$$J^T \Sigma_x J = I, L^T \Sigma_y L = I, \quad (3.25)$$
$$J^T \Sigma_{xy} L = \Sigma = \begin{bmatrix} diag(\sigma_1, \cdots, \sigma_\kappa) & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.26)$$

Definition 3.1 *Given random vectors $y \in \mathcal{R}^m, x \in \mathcal{R}^n$ satisfying (3.21), and let $K = R\Sigma V^T$, J, L be defined in (3.23) and (3.24), respectively. Then,*

$$J_i = \Sigma_x^{-1/2} r_i, L_i = \Sigma_y^{-1/2} v_i, i = 1, \cdots, \kappa, \quad (3.27)$$

are called canonical correlation vectors,

$$\eta_i = J_i^T x, \varphi_i = L_i^T y, i = 1, \cdots, \kappa, \quad (3.28)$$

canonical correlation variables, and $\sigma_1, \cdots, \sigma_\kappa$ are called canonical correlation coefficients.

It is clear that the first l $(\leq \kappa)$ canonical correlation vectors $J_1, L_1, \cdots, J_l, L_l$ define the l mostly correlated linear combinations corresponding to the first l largest canonical correlation coefficients $\sigma_1, \cdots, \sigma_l$. Moreover, it holds for the canonical correlation vectors

$$\bar{J} = \begin{bmatrix} J_1 \cdots J_\kappa \end{bmatrix}, \bar{L} = \begin{bmatrix} L_1 \cdots L_\kappa \end{bmatrix}, \quad (3.29)$$
$$\bar{J}^T \Sigma_x \bar{J} = I \in \mathcal{R}^{\kappa \times \kappa}, \bar{L}^T \Sigma_y \bar{L} = I \in \mathcal{R}^{\kappa \times \kappa},$$
$$\bar{J}^T \Sigma_{xy} \bar{L} = diag(\sigma_1, \cdots, \sigma_\kappa) =: \bar{\Sigma}.$$

3.3.2 Application to Fault Detection and Estimation

The basic idea behind the application of CCA method to fault detection and estimation consists in reducing the process uncertainty by making use of the existing correlation between two measurement vectors. To simplify our study, we assume that

$$\mathcal{E}(x) = 0, \mathcal{E}(y) = 0$$

without loss of generality. For our purpose, define

$$r_1 = \bar{J}^T x - \bar{\Sigma}\bar{L}^T y, r_2 = \bar{L}^T y - \bar{\Sigma}\bar{J}^T x. \tag{3.30}$$

It turns out

$$\begin{aligned}\mathcal{E}\left(r_1 r_1^T\right) &= \bar{J}^T \Sigma_x \bar{J} + \bar{\Sigma}\bar{L}^T \Sigma_y \bar{L}\bar{\Sigma} - \bar{J}^T \Sigma_{xy} \bar{L}\bar{\Sigma} - \bar{\Sigma}\bar{L}^T \Sigma_{yx} \bar{J} \\ &= I - \bar{\Sigma}\bar{\Sigma} = diag\left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2\right), \qquad (3.31) \\ \mathcal{E}\left(r_2 r_2^T\right) &= \bar{L}^T \Sigma_y \bar{L} + \bar{\Sigma}\bar{J}^T \Sigma_x \bar{J}\bar{\Sigma} - \bar{L}^T \Sigma_{yx} \bar{J}\bar{\Sigma} - \bar{\Sigma}\bar{J}^T \Sigma_{xy} \bar{L} \\ &= I - \bar{\Sigma}\bar{\Sigma} = diag\left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2\right). \qquad (3.32)\end{aligned}$$

In general, when we define

$$r_1 = J^T x - \Sigma L^T y, r_2 = L^T y - \Sigma^T J^T x, \tag{3.33}$$

the covariance matrices of r_1, r_2 satisfy

$$\begin{aligned}\mathcal{E}\left(r_1 r_1^T\right) &= J^T \Sigma_x J + \Sigma L^T \Sigma_y L \Sigma^T - J^T \Sigma_{xy} L \Sigma^T - \Sigma L^T \Sigma_{yx} J \\ &= I - \Sigma\Sigma^T = diag\left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2, 1, \cdots, 1\right), \qquad (3.34) \\ \mathcal{E}\left(r_2 r_2^T\right) &= L^T \Sigma_y L + \Sigma^T J^T \Sigma_x J \Sigma - L^T \Sigma_{yx} J \Sigma - \Sigma^T J^T \Sigma_{xy} L \\ &= I - \Sigma^T \Sigma = diag\left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2, 1, \cdots, 1\right). \qquad (3.35)\end{aligned}$$

A comparison with

$$\mathcal{E}\left(J^T x x^T J\right) = J^T \Sigma_x J = I, \mathcal{E}\left(L^T y y^T L\right) = L^T \Sigma_y L = I,$$

which are the normalised covariance matrices of x, y, respectively, makes it clear that the covariance matrix of the random vector r_1 or r_2 under consideration becomes smaller when the correlated measurements are taken into account. In fact, r_1, r_2 can be re-written as

$$r_1 = J^T \left(x - \Sigma_{xy} L L^T y\right) = R^T \Sigma_x^{-1/2} \left(x - \Sigma_{xy} \Sigma_y^{-1} y\right), \tag{3.36}$$

$$r_2 = L^T \left(y - \Sigma_{yx} J J^T x\right) = V^T \Sigma_y^{-1/2} \left(y - \Sigma_{yx} \Sigma_x^{-1} x\right). \tag{3.37}$$

Note that

$$\hat{x} = \Sigma_{xy} \Sigma_y^{-1} y, \hat{y} = \Sigma_{yx} \Sigma_x^{-1} x$$

are LMS estimates for x, y by means of y, x, respectively, thus the estimation errors, $x - \hat{x}, y - \hat{y}$, have the minimum variance. In this regard, r_1, r_2 can be understood as residual signals. This motivates us to use signals r_1, r_2 for fault detection and estimation purpose.

Let

$$y = f_y + \varepsilon_y \in \mathcal{R}^m, \varepsilon_y \sim \mathcal{N}(0, \Sigma_y), \tag{3.38}$$
$$x = f_x + \varepsilon_x \in \mathcal{R}^n, \varepsilon_x \sim \mathcal{N}(0, \Sigma_x) \tag{3.39}$$

be the process models for (sub-)process y, x, where f_y, f_x represent fault vectors in the process measurements y and x, respectively. We assume that f_y and f_x are not present in the processes simultaneously. Suppose that $\varepsilon_y, \varepsilon_x$ are correlated with

$$\mathcal{E}\left(\varepsilon_x \varepsilon_y^T\right) = \Sigma_{xy}.$$

Then, after determining J, L, Σ according to (3.23) and (3.24), we have the following fault detection solutions:

- define the test statistics

$$J_x = r_1^T \left(diag\left(1-\sigma_1^2, \cdots, 1-\sigma_\kappa^2, 1, \cdots, 1\right)\right)^{-1} r_1, \tag{3.40}$$
$$r_1 = J^T x - \Sigma L^T y,$$
$$J_y = r_2^T \left(diag\left(1-\sigma_1^2, \cdots, 1-\sigma_\kappa^2, 1, \cdots, 1\right)\right)^{-1} r_2, \tag{3.41}$$
$$r_2 = L^T y - \Sigma^T J^T x,$$

  where it is assumed that $\sigma_1 < 1$;
- set the thresholds: for a given acceptable FAR α

$$J_{th,x} = \chi_{\alpha,x}, \Pr\left\{\chi^2(n) > \chi_{\alpha,x}\right\} = \alpha, \tag{3.42}$$
$$J_{th,y} = \chi_{\alpha,y}, \Pr\left\{\chi^2(m) > \chi_{\alpha,y}\right\} = \alpha; \tag{3.43}$$

- define the detection logic

$$J_x \leq J_{th,x} \Longrightarrow \text{fault-free, otherwise faulty with } x, \tag{3.44}$$
$$J_y \leq J_{th,y} \Longrightarrow \text{fault-free, otherwise faulty with } y. \tag{3.45}$$

It should be noticed that the above fault detection solutions only allow a successful fault detection, but do not guarantee a perfect fault isolation. This fact can be clearly seen from the following relations

$$r_1 = J^T x - \Sigma L^T y = J^T f_x - \Sigma L^T f_y + J^T \varepsilon_x - \Sigma L^T \varepsilon_y,$$
$$r_2 = L^T y - \Sigma^T J^T x = L^T f_y - \Sigma^T J^T f_x + L^T \varepsilon_y - \Sigma^T J^T \varepsilon_x,$$

which mean that r_1, r_2 will be affected by both f_x, f_y. On the other hand, it holds

$$\begin{cases} \mathcal{E}(J_x) = f_x^T \Sigma_{x,1} f_x + n, \mathcal{E}(J_y) = f_x^T \Sigma_{x,2} f_x + m, \text{ for } f_x \neq 0, f_y = 0, \\ \mathcal{E}(J_y) = f_y^T \Sigma_{y,1} f_y + m, \mathcal{E}(J_x) = f_y^T \Sigma_{y,2} f_y + n, \text{ for } f_y \neq 0, f_x = 0, \end{cases}$$

$$\Sigma_{x,1} = \Sigma_x^{-1/2} R \left(diag \left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2, 1, \cdots, 1 \right) \right)^{-1} R^T \Sigma_x^{-1/2},$$
$$\Sigma_{x,2} = \Sigma_x^{-1/2} R diag \left(\sigma_1^2 \left(1 - \sigma_1^2 \right)^{-1}, \cdots, \sigma_\kappa^2 \left(1 - \sigma_\kappa^2 \right)^{-1}, 0, \cdots, 0 \right) R^T \Sigma_x^{-1/2},$$
$$\Sigma_{y,1} = \Sigma_y^{-1/2} V \left(diag \left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2, 1, \cdots, 1 \right) \right)^{-1} V^T \Sigma_y^{-1/2},$$
$$\Sigma_{y,2} = \Sigma_y^{-1/2} V diag \left(\sigma_1^2 \left(1 - \sigma_1^2 \right)^{-1}, \cdots, \sigma_\kappa^2 \left(1 - \sigma_\kappa^2 \right)^{-1}, 0, \cdots, 0 \right) V^T \Sigma_y^{-1/2}.$$

It turns out, on the assumption $\sigma_1 < 1$,

$$\Sigma_{x,1} > \Sigma_{x,2}, \Sigma_{y,1} > \Sigma_{y,2} \Longrightarrow$$
$$\mathcal{E}(J_x) > \mathcal{E}(J_y) - m + n, \text{ for } f_x \neq 0, f_y = 0, \tag{3.46}$$
$$\mathcal{E}(J_x) < \mathcal{E}(J_y) - m + n, \text{ for } f_y \neq 0, f_x = 0. \tag{3.47}$$

Inequalities (3.46) and (3.47) can be applied as a decision logic for fault isolation. Note that if J_x, J_y are used, instead of their mean, for this purpose, false isolation decisions can be made. The rate of false isolation decision depends on f_x, f_y, which are in general unknown. In order to reduce false isolation decisions, we can collect data and estimate $\mathcal{E}(J_x)$ and $\mathcal{E}(J_y)$.

Following our discussion on the solution of the basic fault detection problem reviewed in Sect. 3.1, it is evident that the above solutions, $\left\{ J_x, J_{th,x} \right\}$ and $\left\{ J_y, J_{th,y} \right\}$, are the optimal solution for detecting faults f_x and f_y, thanks to the fact that $x - \hat{x}, y - \hat{y}$ have the minimum variance. On the other hand, attention should be paid to the assumption that f_y and f_x are not present in the (sub-)processes simultaneously. If this is not the case, then the overall model

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix}, \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix} \right)$$

should be used for the detection purpose, which is the basic fault detection problem discussed in Sect. 3.1.

We now consider another assumption $\sigma_1 < 1$. Suppose that this is not true and

$$\sigma_1 = \sigma_2 = \cdots = \sigma_l = 1, l \leq \kappa, \tag{3.48}$$

which means, according to (3.23) and (3.24),

$$\begin{bmatrix} J_1^T \\ \vdots \\ J_l^T \end{bmatrix} \Sigma_{xy} \begin{bmatrix} L_1 \cdots L_l \end{bmatrix} = I.$$

It turns out, in the fault-free case,

$$\mathcal{E}\left(\left(\begin{bmatrix} J_1^T \\ \vdots \\ J_l^T \end{bmatrix} x - \begin{bmatrix} L_1^T \\ \vdots \\ L_l^T \end{bmatrix} y\right)\left(\begin{bmatrix} J_1^T \\ \vdots \\ J_l^T \end{bmatrix} x - \begin{bmatrix} L_1^T \\ \vdots \\ L_l^T \end{bmatrix} y\right)^T\right) = 0.$$

Recalling

$$\begin{bmatrix} J_1^T & L_1^T \\ \vdots & \vdots \\ J_l^T & L_l^T \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} J_1^T & L_1^T \\ \vdots & \vdots \\ J_l^T & L_l^T \end{bmatrix}\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix}, \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right),$$

it can be concluded that

$$\begin{bmatrix} J_1^T \\ \vdots \\ J_l^T \end{bmatrix} x - \begin{bmatrix} L_1^T \\ \vdots \\ L_l^T \end{bmatrix} y = 0. \tag{3.49}$$

As a result, the original fault detection problem can be re-formulated as two fault detection problems:

- fault detection by means of a plausibility check based on (3.49),
- fault detection using the signals

$$r_1 = \begin{bmatrix} J_{l+1}^T \\ \vdots \\ J_n^T \end{bmatrix} x - \bar{\Sigma} \begin{bmatrix} L_{l+1}^T \\ \vdots \\ L_m^T \end{bmatrix} y, r_2 = \begin{bmatrix} L_{l+1}^T \\ \vdots \\ L_m^T \end{bmatrix} y - \bar{\Sigma}^T \begin{bmatrix} J_{l+1}^T \\ \vdots \\ J_n^T \end{bmatrix} x,$$
$$\bar{\Sigma} = \begin{bmatrix} diag(\sigma_{l+1}, \cdots, \sigma_\kappa) & 0 \\ 0 & 0 \end{bmatrix}.$$

Concerning fault estimation, signals

$$\hat{f}_x = x - \hat{x}, \hat{x} = \Sigma_{xy}\Sigma_y^{-1} y, \hat{f}_y = y - \hat{y}, \hat{y} = \Sigma_{yx}\Sigma_x^{-1} x$$

deliver estimates for f_x, f_y respectively, after a fault, either f_x or f_y, is detected and isolated. They are LMS estimations and satisfy

$$\mathcal{E}\left(\left(f_x - \hat{f}_x\right)\left(f_x - \hat{f}_x\right)^T\right) =$$
$$R^T \Sigma_x^{-1/2}\left(diag\left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2, 1, \cdots, 1\right)\right)^{-1} \Sigma_x^{-1/2} R,$$
$$\mathcal{E}\left(\left(f_y - \hat{f}_y\right)\left(f_y - \hat{f}_y\right)^T\right) =$$
$$V^T \Sigma_y^{-1/2}\left(diag\left(1 - \sigma_1^2, \cdots, 1 - \sigma_\kappa^2, 1, \cdots, 1\right)\right)^{-1} \Sigma_y^{-1/2} V.$$

3.3.3 CCA and GLR

In the previous sub-section, we have claimed that the test statistics,

$$\begin{aligned} &J_x = r_1^T \left(I - \Sigma\Sigma^T\right)^{-1} r_1, J_y = r_2^T \left(I - \Sigma^T\Sigma\right)^{-1} r_2, \\ &r_1 = J^T x - \Sigma L^T y, r_2 = L^T y - \Sigma^T J^T x, \end{aligned}$$

can be applied for detecting the faults f_x, f_y as given in the models (3.38) and (3.39), respectively, and they deliver, on the assumption that f_x, f_y are not present in the process simultaneously, the best fault detectability. In this sub-section, we are going to prove this statement by handling the CCA-based FD as a special case of the GLR-based solution.

In order to simplify the notation and subsequent discussion, we first do a normalisation on x, y in the process model

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix}, \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right)$$

as follows

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} R^T \Sigma_x^{-1/2} & 0 \\ 0 & V^T \Sigma_y^{-1/2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} R^T \Sigma_x^{-1/2} x \\ V^T \Sigma_y^{-1/2} y \end{bmatrix}$$

with R, V as the unitary matrices defined in (3.23). It turns out

$$\begin{aligned} &\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \bar{f}_x \\ \bar{f}_y \end{bmatrix} + \begin{bmatrix} \bar{\varepsilon}_x \\ \bar{\varepsilon}_y \end{bmatrix}, \qquad (3.50) \\ &\begin{bmatrix} \bar{f}_x \\ \bar{f}_y \end{bmatrix} = \begin{bmatrix} R^T \Sigma_x^{-1/2} f_x \\ V^T \Sigma_y^{-1/2} f_y \end{bmatrix}, \begin{bmatrix} \bar{\varepsilon}_x \\ \bar{\varepsilon}_y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} I & \Sigma \\ \Sigma^T & I \end{bmatrix}\right), \end{aligned}$$

where

$$\Sigma = R^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} V,$$

as defined in (3.23). Now, we are in the position to formulate our detection problem, under the assumption that f_x, f_y are not present in the process simultaneously, as the following two FD problems:

- detecting $\bar{f}_x$ in the model

$$z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \bar{f}_x + \begin{bmatrix} \bar{\varepsilon}_x \\ \bar{\varepsilon}_y \end{bmatrix}, \qquad (3.51)$$

- detecting $\bar{f}_y$ in the model

$$z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \bar{f}_y + \begin{bmatrix} \bar{\varepsilon}_x \\ \bar{\varepsilon}_y \end{bmatrix}. \tag{3.52}$$

Next, we focus on solving FD problem for process (3.51) using the standard GLR method. Note that (3.51) is given exactly in the form of (3.10) with

$$E_f = \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

It follows from the discussion in Sect. 3.2 that the test statistic

$$\begin{aligned}
J &= z^T \left(E_f^-\right)^T \left(E_f^- \begin{bmatrix} I & \Sigma \\ \Sigma^T & I \end{bmatrix} \left(E_f^-\right)^T\right)^{-1} E_f^- z \\
&= z^T \begin{bmatrix} I & \Sigma \\ \Sigma^T & I \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left(\begin{bmatrix} I \\ 0 \end{bmatrix}^T \begin{bmatrix} I & \Sigma \\ \Sigma^T & I \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix}^T \begin{bmatrix} I & \Sigma \\ \Sigma^T & I \end{bmatrix}^{-1} z \\
&= (\bar{x} - \Sigma \bar{y})^T \left(I - \Sigma \Sigma^T\right)^{-1} (\bar{x} - \Sigma \bar{y})
\end{aligned}$$

delivers the best solution. Recall that

$$\bar{x} - \Sigma \bar{y} = R^T \Sigma_x^{-1/2} x - \Sigma V^T \Sigma_y^{-1/2} y = R^T \Sigma_x^{-1/2} \left(x - \Sigma_{xy} \Sigma_y^{-1} y\right) = r_1,$$

and covariance matrix of r_1 is $I - \Sigma \Sigma^T$. Hence, it is proved that

$$J_x = r_1^T \left(I - \Sigma \Sigma^T\right)^{-1} r_1$$

is the best test statistic for detecting the faults in x. Analogue to it, it can also be proved that the test statistic J_y with r_2 given in (3.41) is the best statistic for detecting the faults in y.

3.4 Fault Detection and Estimation with Deterministic Disturbances

Comparing with the investigations on fault detection and estimation issues for dynamic processes with deterministic disturbances, few results have been reported on dealing with the similar topics for static processes. In the present section, we will review some basic methods for the latter case.

3.4.1 A Basic Fault Detection Problem

We first consider a basic fault detection problem and suppose that the process model is given as follows

$$y = f + E_d d \in \mathcal{R}^m, m \geq 1, \tag{3.53}$$

where d is unknown disturbance vector and norm-bounded,

$$d \in \mathcal{R}^{k_d}, \|d\|^2 = d^T d \leq \delta_d^2 \tag{3.54}$$

with known δ_d^2, the known matrix E_d satisfies

$$rank\left(E_d\right) = m, \tag{3.55}$$

and fault vector f is as defined in (3.2). The following theorem gives an optimal solution $\{J, J_{th}\}$ for the fault detection problem defined in Definition 2.7.

Theorem 3.1 *Given model (3.53), then*

$$J = y^T \left(E_d E_d^T\right)^{-1} y, J_{th} = \delta_d^2 \tag{3.56}$$

are a solution of the fault detection problem defined in Definition 2.7.

Proof Following Theorem 2.1, we just need to prove that

$$\mathcal{M}_d^- = \left(E_d E_d^T\right)^{-1/2}$$

satisfies conditions (2.21)–(2.22). Note that

$$rank\left(\left(E_d E_d^T\right)^{-1/2} E_d\right) = m, \sigma_i\left(\left(E_d E_d^T\right)^{-1/2} E_d\right) = 1$$

with $i = 1, \cdots, m$. This ensures

$$\forall d, d^T E_d^T \left(E_d E_d^T\right)^{-1} E_d d \leq d^T d \Longrightarrow \text{(2.21) is true,}$$
$$\forall f, \exists z, \text{ s.t. } f^T \left(E_d E_d^T\right)^{-1} f = z^T E_d^T \left(E_d E_d^T\right)^{-1} E_d z.$$

Consider f satisfying

$$\left\|\mathcal{M}_d^- \mathcal{M}_f f\right\|^2 = \left\|\left(E_d E_d^T\right)^{-1/2} f\right\|^2 \leq \delta_d^2.$$

It is evident that

$$\begin{gathered}d = E_d^T \left(E_d E_d^T\right)^{-1} f \Longrightarrow \\ \mathcal{M}_d^- \mathcal{M}_d d = \left(E_d E_d^T\right)^{-1/2} f = \mathcal{M}_d^- \mathcal{M}_f f,\end{gathered}$$

which proves that (2.22) holds.

Example 3.1 *Note that the result in Theorem 3.1 is different from the solution given in Example 2.1, although model (3.53) is indeed a special case of the process model considered in Example 2.1. In this example, we demonstrate that both solutions are equivalent in regard of solving the defined optimal FD problem. In fact, this can be well recognised by noting the fact that the evaluation functions and the corresponding thresholds in both solutions are identical, i.e.*

$$\begin{gathered}M_d^T \left(M_d M_d^T\right)^{-1} = E_d^T \left(E_d E_d^T\right)^{-1}, M_d = E_d \Longrightarrow \\ y^T \left(M_d M_d^T\right)^{-1} M_d M_d^T \left(M_d M_d^T\right)^{-1} y = y^T \left(E_d E_d^T\right)^{-1} y, \\ J_{th} = \delta_d^2.\end{gathered}$$

Notice that $E_d^T \left(E_d E_d^T\right)^{-1}$ *can be written as*

$$E_d^T \left(E_d E_d^T\right)^{-1} = E_d^T \left(E_d E_d^T\right)^{-1/2} \left(E_d E_d^T\right)^{-1/2},$$

and

$$\left(E_d^T \left(E_d E_d^T\right)^{-1/2}\right)^T E_d^T \left(E_d E_d^T\right)^{-1/2} = I.$$

This explains why the both solutions are identical.

Motivated by this example, we would like to call reader's attention to an alternative interpretation of the above solution, which is useful for our subsequent study on fault detection in dynamic processes. Given model (3.53), in the fault-free case,

$$\hat{d} = E_d^- y, E_d^- = E_d^T \left(E_d E_d^T\right)^{-1}, \tag{3.57}$$

is an LS estimation for d. Since all available information about d is its norm-boundedness given in (3.54), it is reasonable to apply the norm of the LS estimate of d,

$$\left\|\hat{d}\right\|^2 = y^T \left(E_d^-\right)^T E_d^- y = y^T \left(E_d E_d^T\right)^{-1} y,$$

to build the evaluation function and to compare it with the known norm-boundedness δ_d^2 (as threshold). As demonstrated in Theorem 3.1, in this manner we can find the optimal solution of the fault detection problem as well.

The solution for the fault detection problem defined in Definition 2.9 is evidently given by

$$J = \|y\|^2, J_{th} = \beta^2 \tag{3.58}$$

with the given margin of detectable faults β. The proof of this result is straightforward. Considering that for process (3.53)

$$\begin{gathered}\mathcal{D}_{f,\beta} = \left\{ f \left| f \in \mathcal{D}_f, \|f\| \leq \beta \right. \right\}, \mathcal{I}_{f,\beta} = \left\{ y_f \left| y_f = f, f \in \mathcal{D}_{f,\beta} \right. \right\}, \\ \mathcal{I}_{d,\beta} = \left\{ y_d \left| y_d = E_d d \in \mathcal{I}_{f,\beta}, d \in \mathcal{D}_d \right. \right\},\end{gathered}$$

it is ensured that

$$\begin{gathered}\forall f \notin \mathcal{D}_{f,\beta}, d = 0, J(y) - J_{th} = \|f\|^2 - \beta^2 > 0, \\ \forall y \in \mathcal{I}_{d,\beta}, f = 0, J(y) - J_{th} = \|E_d d\|^2 - \beta^2 \leq 0.\end{gathered}$$

In fact, by means of Corollary 2.1 this result can be directly proved. Note that on the assumption of process model (3.53), y is indeed an LS estimation of f. This means, the norm of the LS estimation of f can serve as the solution of the fault detection problem defined in Definition 2.9.

3.4.2 A General Form of Fault Detection and Estimation

Analogue to the statistic case, we now consider the extended model form

$$y = E_f f + E_d d \in \mathcal{R}^m, \tag{3.59}$$

where E_f, E_d are as defined in (3.11) and (3.55), respectively. It holds

$$\begin{gathered}\mathcal{I}_d = \left\{ y_d \left| y_d = E_d d, \forall d \in \mathcal{R}^{k_d}, \|d\| \leq \delta_d \right. \right\}, \\ \mathcal{D}_{f,undetc} = \left\{ f \left| y_f = E_f f \in \mathcal{I}_d \right. \right\}.\end{gathered}$$

Let E_f^- be the left inverse of E_f given by

$$E_f^- = \left(E_f^T \left(E_d E_d^T \right)^{-1} E_f \right)^{-1} E_f^T \left(E_d E_d^T \right)^{-1}, E_f^- E_f = I.$$

It holds

$$\bar{y} = E_f^- y = f + E_f^- E_d d \in \mathcal{R}^{k_f}. \tag{3.60}$$

As a result, $\mathcal{D}_{f,undetc}$ can be re-written as

$$\begin{gathered}\mathcal{D}_{f,undetc} = \left\{ f \left| \bar{y}_f = f \in \mathcal{I}_{d,\bar{y}} \right. \right\}, \\ \mathcal{I}_{d,\bar{y}} = \left\{ \bar{y}_d \left| \bar{y}_d = E_f^- E_d d, \forall d \in \mathcal{R}^{k_d}, \|d\| \leq \delta_d \right. \right\}.\end{gathered}$$

The solutions to the fault detection problems defined in Definition 2.7 and Definition 2.9 are respectively given in the following two theorems.

Theorem 3.2 *Given model (3.59), then*

$$J = y^T \left(E_d E_d^T\right)^{-1} M \left(E_d E_d^T\right)^{-1} y,\ J_{th} = \delta_d^2, \tag{3.61}$$
$$M = E_f \left(E_f^T \left(E_d E_d^T\right)^{-1} E_f\right)^{-1} E_f^T,$$

are a solution of the fault detection problem defined in Definition 2.7.

Proof Considering that by the transformation $\bar{y} = E_f^- y$, condition (2.27) holds, i.e.

$$rank\left(E_f^- E_d\right) = k_f = \dim\left(\bar{y}\right),$$

the fault detection problems with process model (3.59) becomes equivalent to the one with model (3.53). Note further

$$\left(E_d E_d^T\right)^{-1} M \left(E_d E_d^T\right)^{-1} = \left(E_f^-\right)^T E_f^T \left(E_d E_d^T\right)^{-1} E_f E_f^-,$$
$$E_f^- E_d E_d^T \left(E_f^-\right)^T = \left(E_f^T \left(E_d E_d^T\right)^{-1} E_f\right)^{-1}.$$

We have

$$J = \bar{y}^T \left(E_f^- E_d E_d^T \left(E_f^-\right)^T\right)^{-1} \bar{y}.$$

Thus, the solution (3.61), following Theorem 3.1, is a solution of the fault detection problem defined in Definition 2.7.

For the same argument and following the discussion in the end of the last sub-section, we have also the following theorem.

Theorem 3.3 *Given model (3.59) and margin of detectable faults β, then*

$$J = \|\bar{y}\|^2 = \left\|E_f^- y\right\|^2,\ J_{th} = \beta^2 \tag{3.62}$$

are a solution of the fault detection problem defined in Definition 2.9.

It is worth remarking that $\hat{f} = E_f^- y$ is an LS estimate of f and can be used for the fault estimation purpose. On the other hand, it should be pointed out that the use of $\hat{f}^T \hat{f}$ as an evaluation function for the detection purpose delivers poor detectability performance, since it is not optimal for the fault detection problem defined in Definition 2.7. This can be seen from the following discussion as well. For

$$J = \hat{f}^T \hat{f} = y^T \left(E_f^-\right)^T E_f^- y,$$

it holds, in the fault-free case,

$$J = d^T E_d^T \left(E_f^-\right)^T E_f^- E_d d,$$

which requires the threshold setting

$$J_{th} = \bar{\sigma}^2 \left(E_f^- E_d\right) \delta_d^2,$$

in order to guarantee condition (2.14). On the other hand, since for $d = 0$

$$J = \hat{f}^T \hat{f} = f^T f \Longrightarrow J - J_{th} = f^T f - \bar{\sigma}^2 \left(E_f^- E_d\right) \delta_d^2,$$

and considering (3.60), it becomes evident that

$$\exists f \notin \mathcal{D}_{f,undetc}, \text{ s.t. } J - J_{th} \leq 0.$$

That is, f is not detectable, as far as $E_f^- E_d \neq I$.

3.5 The Data-Driven Solutions of the Detection and Estimation Problems

In the previous sections, fault detection and estimation problems have been handled on the assumption that a model exists for the process under consideration. In the real application world, this is often not the case. This motivates the development of data-driven fault detection and estimation methods. In this section, we briefly introduce the data-driven solutions of the fault detection and estimation problems handled in the last four sections, and then review the three well-established basic data-driven fault detection methods: PCA (principal component analysis), PLS (partial least squares) and CCA.

In the context of data-driven fault detection and estimation formulation, it is often assumed that the model structure/form is given, but the model parameters are unknown *a prior*. On the other hand, a huge number of historic data, $y_1, \cdots y_N$, $N >> 1$, are available. This motivates the identification of the model parameters using the available data or the integration of the model parameter identification into the fault detection and estimation procedure. In this manner, the fault detection and estimation problems are often solved in a two-step procedure:

- identification of the model parameters using the recorded data. This is also called training phase and runs typically offline,
- application of an existing (model-based) fault detection and estimation scheme.

3.5.1 Fault Detection and Estimation in Statistic Processes with sufficient Training Data

It is clear that for the application of the fault detection and estimation methods for statistic processes introduced in the previous sections, the mean vector and covariance matrix of y, $\mathcal{E}(y)$, $var(y)$, are needed. It is well-known that

$$\bar{y}(N) = \frac{1}{N}\sum_{i=1}^{N} y_i \text{ and } S_{N-1} = \frac{1}{N-1}\sum_{i=1}^{N} (y_i - \bar{y}(N))(y_i - \bar{y}(N))^T \tag{3.63}$$

are sample mean vector and covariance matrix of y, and for $N \to \infty$

$$\lim_{N\to\infty} \bar{y}(N) = \mathcal{E}(y) = \mu \text{ and } \lim_{N\to\infty} S_{N-1} = var(y) = \Sigma. \tag{3.64}$$

On the assumption that N is sufficiently large so that

$$\bar{y}(N) \approx \mu, S_{N-1} \approx \Sigma,$$

the following procedure can be used for dealing with the fault detection and estimation problems:

- Offline training: computation of

$$\bar{y}(N) = \frac{1}{N}\sum_{i=1}^{N} y_i, S_{N-1} = \frac{1}{N-1}\sum_{i=1}^{N} (y_i - \bar{y}(N))(y_i - \bar{y}(N))^T ;$$

- Offline design: setting thresholds using the given formulas;
- Online fault detection and estimation using the given test statistics and estimation algorithms.

3.5.2 Fault Detection Using Hotelling's T^2 test statistic

In practical cases, the number of data is often limited. This requires special handling of the fault detection solutions. For our purpose, consider model

$$y = f + \varepsilon \in \mathcal{R}^m, \varepsilon \sim \mathcal{N}(\mu, \Sigma)$$

with unknown μ, Σ. Assume that there are two groups of process data available: training data (recorded for the offline computation) $y_i, i = 1, \cdots, N$, online measurement data $y_{k+i}, i = 1, \cdots, n$, and $N >> n$. Note that in our previous study,

$n = 1$. Recall that the training data should be recorded in the fault-free case, that means the data (samples) have been generated on the model assumption

$$y = \varepsilon \in \mathcal{R}^m, \mathcal{E}(y) = \mathcal{E}(\varepsilon) = \mu. \tag{3.65}$$

Differently, the online measurement data may cover both the fault-free or faulty cases. Hence, the model assumption is

$$y = f + \varepsilon \in \mathcal{R}^m, \mathcal{E}(y) = \mu + f =: \mu_f, f = \begin{cases} 0, \text{fault-free,} \\ \text{constant} \neq 0, \text{ faulty.} \end{cases} \tag{3.66}$$

In the context of a fault detection, we are now able to re-formulate our original problem as

$$\begin{cases} \mu - \mu_f = 0 \Longrightarrow \text{ fault-free,} \\ \mu - \mu_f \neq 0 \Longrightarrow \text{ faulty.} \end{cases} \tag{3.67}$$

For our purpose, the (possible) difference between the means of the two data sets should be checked. To this end, consider

$$\bar{y}(N) - \bar{y} \sim \mathcal{N}\left(f, \frac{N+n}{nN}\Sigma\right), \bar{y}(N) = \frac{1}{N}\sum_{i=1}^{N} y_i, \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_{k+i}. \tag{3.68}$$

Since Σ is unknown, it will be estimated by offline and online data sets, respectively. Let

$$S_{off} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \bar{y}(N)\right)\left(y_i - \bar{y}(N)\right)^T, S_{on} = \frac{1}{n}\sum_{i=1}^{n}\left(y_{k+i} - \bar{y}\right)\left(y_{k+i} - \bar{y}\right)^T$$

be sample covariance matrices of the two data sets and

$$\begin{aligned} S &= \frac{NS_{off} + nS_{on}}{n+N} \\ &= \frac{1}{n+N}\left(\sum_{i=1}^{N}\left(y_i - \bar{y}(N)\right)\left(y_i - \bar{y}(N)\right)^T + \sum_{i=1}^{n}\left(y_{k+i} - \bar{y}\right)\left(y_{k+i} - \bar{y}\right)^T\right). \end{aligned} \tag{3.69}$$

The following theorem plays a central role for building the test statistic and setting the threshold.

Theorem 3.4 *Let* $f, \bar{y}(N), \bar{y}, S$ *be defined in (3.66), (3.68) and (3.69). It holds*

$$\frac{nN(n+N-2)}{(n+N)^2}\left(\bar{y}(N) - \bar{y}\right)^T S^{-1}\left(\bar{y}(N) - \bar{y}\right) \sim \mathcal{T}^2(m, n+N-2), \tag{3.70}$$

where $\mathcal{T}^2(m, n+N-2)$ *is Hotelling* $\mathcal{T}^2$*-distribution with* m *and* $(n+N-2)$ *degrees of freedom.*

The above theorem is a standard result on the Hotelling $\mathcal{T}^2$ -distribution and its proof can be found in books on multivariate analysis. The reader is referred to the references given at the end of this chapter.

Consider further

$$\mathcal{T}^2(m, n+N-2) = \frac{m(n+N-2)}{n+N-m-1}\mathcal{F}(m, n+N-m-1),$$

where $\mathcal{F}(m, n+N-2)$ denotes $\mathcal{F}$-distribution with m and $(n+N-1)$ degrees of freedom. As a result, for the test statistic defined by

$$\left(\bar{y}(N)-\bar{y}\right)^T S^{-1}\left(\bar{y}(N)-\bar{y}\right), \tag{3.71}$$

the corresponding threshold is set to be

$$J_{th} = \frac{m\,(n+N)^2}{nN\,(n+N-m-1)}\mathcal{F}_\alpha(m, n+N-m-1), \tag{3.72}$$

for a given acceptable $FAR\ \alpha$.

Note that S consists of two terms: $\frac{NS_{off}}{n+N}, \frac{nS_{on}}{n+N}$. For $n=1$, it holds

$$S = \frac{NS_{off}}{N+1} = \frac{N-1}{N+1}\frac{1}{N-1}\sum_{i=1}^{N}\left(y_i-\bar{y}(N)\right)\left(y_i-\bar{y}(N)\right)^T =: \frac{N-1}{N+1}\hat{\Sigma},$$

where

$$\hat{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N}\left(y_i-\bar{y}(N)\right)\left(y_i-\bar{y}(N)\right)^T$$

is a unbiased estimate of Σ. Finally, we define the test statistic

$$J = \left(\bar{y}(N)-y_k\right)^T \hat{\Sigma}^{-1}\left(\bar{y}(N)-y_k\right) = \frac{N-1}{N+1}\left(\bar{y}(N)-y_k\right)^T S^{-1}\left(\bar{y}(N)-y_k\right) \tag{3.73}$$

and the corresponding threshold

$$J_{T^2,th} = \frac{m\left(N^2-1\right)}{N\,(N-m)}\mathcal{F}_\alpha(m, N-m). \tag{3.74}$$

3.5.3 Fault Detection Using Q Statistic

In the T^2 test statistic (3.73), computation of the inverse matrix of $\hat{\Sigma}$ is necessary. By a high dimensional and often ill-conditional $\hat{\Sigma}$, such a computation may cause numerical trouble in practical applications. As an alternative statistic,

$$Q = y^T y \tag{3.75}$$

is widely accepted in practice and applied in the multivariate analysis technique. On the assumption of $y \sim \mathcal{N}(0, \Sigma_y)$, it is proved that the distribution of Q can be approximated by

$$Q \sim g\chi^2(h), \tag{3.76}$$

where $\chi^2(h)$ denotes the χ^2 distribution with h degrees of freedom, and

$$g = \frac{S}{2\mathcal{E}(Q)}, h = \frac{2\mathcal{E}^2(Q)}{S}, S = \mathcal{E}\left((Q - \mathcal{E}(Q))^2\right) = \mathcal{E}\left(Q^2\right) - \mathcal{E}^2(Q). \tag{3.77}$$

For our purpose of fault detection, $\mathcal{E}(Q)$, S can be estimated using the training data, and the threshold setting for the statistic (3.76) is given by

$$J_{th,Q} = g\chi^2_\alpha(h), \tag{3.78}$$

where α is the acceptable FAR.

3.5.4 Application of Principal Component Analysis to Fault Diagnosis

PCA is a basic engineering method and has been successfully used in numerous areas including data compression, feature extraction, image processing, pattern recognition, signal analysis and process monitoring. Thanks to its simplicity and efficiency in processing huge amount of process data, PCA is recognised as a powerful tool of statistical process monitoring and widely used in the process industry for fault detection and diagnosis. In research, PCA often serves as a basic technique for the development of advanced process monitoring and fault diagnosis techniques like recursive or adaptive PCA and kernel PCA.

From χ^2-test statistic to PCA

Recall that χ^2-test statistic (3.4) can be, by an SVD of the covariance matrix Σ, written in form of (3.8). Let

$$P_1 = \left[\, p_1 \cdots p_l \,\right], P_2 = \left[\, p_{l+1} \cdots p_m \,\right].$$

This test statistic can be further written as

$$\begin{aligned} J &= y^T \Sigma^{-1} y \\ &= y^T P_1 diag\left(\sigma_1^{-2}, \cdots, \sigma_l^{-2}\right) P_1^T y + y^T P_2 diag\left(\sigma_{l+1}^{-2}, \cdots, \sigma_m^{-2}\right) P_2^T y. \end{aligned} \quad (3.79)$$

Alternatively, we can define two test statistics as

$$J_1 = y^T P_1 diag\left(\sigma_1^{-2}, \cdots, \sigma_l^{-2}\right) P_1^T y, \quad (3.80)$$

$$J_2 = y^T P_2 diag\left(\sigma_{l+1}^{-2}, \cdots, \sigma_m^{-2}\right) P_2^T y. \quad (3.81)$$

In case that Σ is unknown, but can be estimated using data, we are able to build the above two test statistics by means of (i) a data normalisation and (ii) the estimated Σ, as given in (3.63). This leads to two T^2 test statistics. If, moreover, $\sigma_{l+1}, \cdots, \sigma_m$ are very small and computation of $\sigma_{l+1}^{-2}, \cdots, \sigma_m^{-2}$ would cause numerical problems, the second T^2 test statistic can be substituted by Q statistic introduced in the last sub-section. As a result, we now have the basic form of PCA method.

The basic form of PCA

PCA method is generally applied to solving fault detection problems in static (statistical) processes. The basic PCA algorithms can be summarised as follows.

Algorithm 3.1 *Offline computation (training): Given data $y_i, i = 1, \cdots, N$,*

- *Center the data*

$$\bar{y}(N) = \frac{1}{N} \sum_{i=1}^{N} y_i, \bar{y}_i = y_i - \bar{y}(N) \quad (3.82)$$

  and form the data matrix

$$Y_N = \left[\, \bar{y}_1 \cdots \bar{y}_N \,\right] \in \mathcal{R}^{m \times N}; \quad (3.83)$$

- *Compute the estimation of Σ*

$$\hat{\Sigma} = \frac{1}{N-1} Y_N Y_N^T; \quad (3.84)$$

- *Do an SVD of $\hat{\Sigma}$*

$$\hat{\Sigma} = P \Lambda P^T, \Lambda = diag\left(\sigma_1^2, \cdots, \sigma_m^2\right), \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_m^2; \quad (3.85)$$

- *Determine the number of principal components (PCs) l and decompose P, Λ into*

$$\Lambda = \begin{bmatrix} \Lambda_{pc} & 0 \\ 0 & \Lambda_{res} \end{bmatrix}, \Lambda_{pc} = diag\left(\sigma_1^2, \cdots, \sigma_l^2\right), \tag{3.86}$$

$$\Lambda_{res} = diag\left(\sigma_{l+1}^2, \cdots, \sigma_m^2\right) \in \mathcal{R}^{(m-l)\times(m-l)}, \sigma_l^2 >> \sigma_{l+1}^2,$$

$$P = \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix} \in \mathcal{R}^{m\times m}, P_{pc} \in \mathcal{R}^{m\times l}; \tag{3.87}$$

- *Set two thresholds,*

$$SPE : J_{th,SPE} = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0}, \tag{3.88}$$

$$T_{PCA}^2 : J_{th,T_{PCA}^2} = \frac{l\left(N^2 - 1\right)}{N(N-l)} F_\alpha(l, N-l), \tag{3.89}$$

$$\theta_i = \sum_{j=l+1}^{m} \left(\sigma_j^2\right)^i, i = 1, 2, 3, h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2},$$

for a (given) significance level α with c_α being the normal deviate.

Algorithm 3.2 *Online detection algorithm*

- *Center the received data and denote them by y;*
- *Compute the test statistics*

$$T_{PCA}^2 = y^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T y, \tag{3.90}$$

$$SPE = y^T \left(I - P_{pc} P_{pc}^T\right) y = y^T P_{res} P_{res}^T y, \tag{3.91}$$

where SPE stands for squared prediction error;

- *Make a decision according to the detection logic*

$$SPE \le J_{th,SPE} \text{ and } T_{PCA}^2 \le J_{th,T_{PCA}^2} \Longrightarrow \textit{fault-free, otherwise faulty.}$$

Basic ideas and properties

The original idea behind the PCA is to reduce the dimension of a data set, while retaining, as much as possible, the variation present in the data set. The realisation of this idea can be clearly seen from the decomposition of P, Λ into P_{pc}, P_{res} as well as $\Lambda_{pc}, \Lambda_{res}$, which results in two subspaces in the m-dimensional measurement subspace. The subspace spanned by P_{pc}^T is called principal subspace , which is constructed by those eigenvectors corresponding to the larger singular values, $\sigma_1^2, \cdots, \sigma_l^2$, of the covariance matrix of the normalised measurement vector. That means, the projection

$$\hat{y} = P_{pc}^T y \in \mathcal{R}^l$$

delivers a (much) lower dimensional vector $\hat{y}$ whose covariance matrix is

$$\mathcal{E}\left(P_{pc}^T y y^T P_{pc}\right) = P_{pc}^T P \Lambda P^T P_{pc} = \Lambda_{pc}.$$

$\hat{y}$ retains the major (principal) variation and thus can be viewed as the information carrier. In against, the covariance matrix of the projection onto the residual subspace,

$$y_{res} = P_{res}^T y \in \mathcal{R}^{m-l}, \mathcal{E}\left(y_{res} y_{res}^T\right) = \Lambda_{res}$$

is (significantly) smaller and thus y_{res} can be, in view of its information content, neglected.

For the fault detection purpose, two test statistics, T_{PCA}^2 and SPE, are defined, which are formed by means of the projections

$$\hat{y} = P_{pc}^T y \text{ and } y_{res} = P_{res}^T y,$$

respectively. We would like to call reader's attention to the following aspects:

- Assumption and problem formulation: Although the distribution of the measurement vector is often not explicitly mentioned, it becomes evident from the test statistics and their threshold setting that the application of the PCA technique to fault detection is based on the assumption of the normally distributed measurement vector;
- Estimation of the covariance matrix: In most studies, less attention has been paid to the data centering and normalisation. This step delivers an estimation of the (normalised) covariance matrix Σ. As we have discussed in Sub-section 3.5.1, in the data-driven fault detection framework, this step plays a central role;
- SVD and inverse of the covariance matrix: The SVD of the (estimated) covariance matrix is the core of the PCA technique. It serves, as discussed at the beginning of this sub-section, as a numerical solution for the inverse computation of the covariance matrix;
- SPE and residual subspace: Note that the columns of P_{res}, $p_{l+1}, \cdots, p_m$, span a subspace corresponding to the $m - l$ smallest singular values, $\sigma_{l+1}^2, \cdots, \sigma_m^2$. As discussed in Sect. 3.1 and revealed by (3.9), in this subspace the fault detectability is higher. On the other hand, it should be noticed that SPE is not a most powerful test, since it is not an LR-type statistic, as will be discussed below.

Variation forms of PCA

It is clear that if Λ_{res} is well-conditioning and computing Λ_{res}^{-1} does not cause any numerical problem, it is suggested to apply the test statistic

$$T_H^2 = y^T P_{res} \Lambda_{res}^{-1} P_{res}^T y$$

for the detection purpose. T_H^2 is called Hawkin's T_H^2 statistic and, as discussed in Sect. 3.1, delivers the best fault detectability. Recalling that

$$T_H^2 = y^T P_{res} \Lambda_{res}^{-1} P_{res}^T y \sim \chi^2 (m - l),$$

the corresponding threshold, different from SPE, can be exactly determined using the available χ^2 data table.

In the context of probabilistic PCA (PPCA), the process model under consideration is described by

$$y = Ex + \varepsilon \in \mathcal{R}^m, x \in \mathcal{R}^n, m > n, \tag{3.92}$$

$$\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I), x \sim \mathcal{N}(0, I), \tag{3.93}$$

where Ex represents the process noise with

$$rank\,(E) = n$$

and ε the measurement noise. The major advantage of the PPCA model is the separate modelling of the process and sensor noises, which enables an effective description of the correlations among the process variables and thus increases fault detection performance. On the other hand, such a modelling scheme requires sophisticated modelling algorithms. The so-called expectation and maximisation (EM) algorithm is widely applied for this purpose.

Suppose that we are only interested in detecting process fault modelled by

$$y = E\,(x + f) + \varepsilon \tag{3.94}$$

with fault vector f to be detected. Note that in the fault-free case,

$$\mathcal{E}\left(yy^T\right) = EE^T + \sigma_\varepsilon^2 I.$$

We assume that

$$\min \{\sigma_i\,(E) \neq 0, i = 1, \cdots, n\} > \sigma_\varepsilon.$$

Considering that

$$EE^T + \sigma_\varepsilon^2 I = P \begin{bmatrix} \Lambda_{pc} & 0 \\ 0 & \sigma_\varepsilon^2 I \end{bmatrix} P^T,$$
$$P = \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix}$$

with

$$\Lambda_{pc} = diag\left(\sigma_1^2, \cdots, \sigma_n^2\right), \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2 >> \sigma_\varepsilon^2,$$

fault detection can be achieved by means of the PCA method and using T^2 test statistic

$$T_{PCA}^2 = y^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T y.$$

We would like to call reader's attention to the similarity of process model (3.94) to the model (3.10) studied in Sect. 3.2 . In fact, (3.94) can be re-written as

$$y = E_f f + \bar{\varepsilon}, \bar{\varepsilon} = Ex + \varepsilon \sim \mathcal{N}(0, EE^T + \sigma_\varepsilon^2 I), E_f = E. \tag{3.95}$$

In case that σ_ε^2 is, in comparison with EE^T, sufficiently small, that is

$$EE^T + \sigma_\varepsilon^2 I \approx EE^T,$$

it turns out

$$y^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T y \approx y^T \left(E^-\right)^T \left(E^- \Sigma \left(E^-\right)^T\right)^{-1} E^- y,$$

where the right side of the above equation is the test statistic given in Sect. 3.2 for fault detection under the use of process model (3.10) with

$$\Sigma = EE^T + \sigma_\varepsilon^2 I \approx EE^T.$$

Hence, we can claim that, according to our study in Sect. 3.2 , the test statistic T_{PCA}^2 results in the best fault detection performance. On the other hand, if σ_ε^2 cannot be neglected, it is evident that

$$y^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T y \neq y^T \left(E^-\right)^T \left(E^- \Sigma \left(E^-\right)^T\right)^{-1} E^- y.$$

This means, the PCA method can only deliver sub-optimal fault detection performance. In Chap. 13, we will discuss this detection problem in more details.

3.5.5 LS, PLS and CCA

Roughly speaking, LS and PLS regressions are multivariate analysis methods that build a linear regression between two data sets expressed in form of data matrices. In typical process monitoring applications, LS and PLS regressions are applied to predict key process variables using process measurement data. Key process variables often serve as indicators for process operation performance or product quality, but may be online unavailable or sampled with a long sampling period.

The objective of LS and PLS

Suppose that $y \in \mathcal{R}^m$ and $\theta \in \mathcal{R}^\kappa$ represent the (centered) process measurement vector and key process variable vector, respectively, and

$$y \sim \mathcal{N}(0, \Sigma_y), \theta \sim \mathcal{N}(0, \Sigma_\theta).$$

Consider the following regression model

$$\theta = \Psi y + \varepsilon_\theta \tag{3.96}$$

with $rank(\Psi) = \kappa$, where ε_θ is the part in θ which is uncorrelated with y, that is $\mathcal{E}(\varepsilon_\theta y^T) = 0$. Suppose that Ψ is unknown and will be estimated using available data, $y_1, \cdots, y_N, \theta_1, \cdots, \theta_N$. Let

$$Y_N = \begin{bmatrix} y_1 \cdots y_N \end{bmatrix}, \Theta_N = \begin{bmatrix} \theta_1 \cdots \theta_N \end{bmatrix}.$$

On the assumption that data sets Y_N, Θ_N with a large N are available and $Y_N Y_N^T$ is invertible,

$$\hat{\Psi} = \Theta_N Y_N^T \left(Y_N Y_N^T\right)^{-1}, \hat{\theta} = \hat{\Psi} y \tag{3.97}$$

deliver approximated LS (LMS as well) solutions for Ψ and θ estimations. Noting that

$$\frac{1}{N-1}\Theta_N Y_N^T \approx \mathcal{E}(\theta y^T), \frac{1}{N-1} Y_N Y_N^T \approx \mathcal{E}(y y^T),$$

$\hat{\Psi}$ can also be written into

$$\hat{\Psi} = \frac{1}{N-1}\Theta_N Y_N^T \left(\frac{1}{N-1} Y_N Y_N^T\right)^{-1} \approx \mathcal{E}(\theta y^T)\left(\mathcal{E}(y y^T)\right)^{-1}.$$

In case that

$$rank\left(Y_N Y_N^T\right) < m,$$

the LS estimation for Ψ is given by

$$\hat{\Psi} = \Theta Y_N^T \left(Y_N Y_N^T\right)^+$$

with $\left(Y_N Y_N^T\right)^+$ being the pseudo-inverse of $Y_N Y_N^T$, which can be expressed in terms of the SVD of $Y_N Y_N^T$ as follows

$$Y_N Y_N^T = U \Sigma U^T, \Sigma = \begin{bmatrix} diag(\sigma_1, \cdots, \sigma_l) & 0 \\ 0 & 0 \end{bmatrix}, U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$$
$$\Longrightarrow \left(Y_N Y_N^T\right)^+ = U \Sigma^+ U^T = U_1 diag\left(\sigma_1^{-1}, \cdots, \sigma_l^{-1}\right) U_1^T.$$

The online predicted value $\hat{\theta} = \hat{\Psi} y$ can then be applied for the detection and monitoring purpose.

Note that for a process with a great number of sensors, the computation demand for $\Theta_N Y_N^T \left(Y_N Y_N^T\right)^{-1}$ is considerably high. In addition, computing $\left(Y_N Y_N^T\right)^{-1}$ may cause numerical problems. The PLS algorithms provide alternative solutions for identifying Ψ, in which the solution is approached by iterative computations of a series of optimisation problems. The core of these algorithms consists in the computation of the "mostly correlated" eigenvectors of matrix $Y_N \Theta_N^T$. These algorithms are numerically stable, reliable, and can be applied to highly dimensional processes. Due to their complexity, these algorithms are not included here. The reader is referred to the publications given in References. Instead, we would like to make some remarks.

Remarks on PLS-based fault diagnosis

From the viewpoint of fault diagnosis, the application of PLS regression technique serves establishing a linear regression between two groups of variables using the collected (training) data. The main goal is to predict the process variables based on the established linear regression model and, using the predicted value, to realise process monitoring and to detect changes in the process variables.

Identifying a linear regression model using process data is a trivial engineering problem. There are numerous mathematical and engineering methods for solving this problem. The questions may arise: why and under which conditions should the PLS-based methods be applied? In order to answer these questions, we first summarise the major features of the PLS regression:

- low computation demands: the major computation of a PLS algorithm is the solution of an optimisation problem. It deals with an eigenvalue-eigenvector computation of a $\kappa \times \kappa$-dimensional matrix. Since $\kappa << m$, the computation in each iteration is numerically highly reliable and its cost is low;
- recursive computation: the overall solution is approached step by step. In this manner, numerical stability is achieved and unnecessary computations can be avoided;
- sub-optimal linear regression: PLS regression technique leads to a linear regression which is, in general, sub-optimal in the sense of least squared prediction error.

In this regard, the basic criteria for selecting the PLS regression for fault diagnosis purpose are:

- a great number of measurement variables are available, that is a large m,
- the application goal is to find out a limited number of combinations of the measurement variables, which deliver sufficient information for a reliable prediction of the process variables under monitoring,
- the search procedure can deliver alternative solutions for testing or simulations,
- the computation capacity and resources are limited and there may exist numerical stability and reliability concerns.

It should be pointed out that nowadays the last point is less critical in engineering applications, although it was, a couple of decades ago, a convincing argument for the application of PLS instead of, for instance, LS method.

On LS, PLS and CCA

In Sub-section 3.3.2, a CCA-based fault detection approach has been introduced, in which the CCA method is applied to the computation of an LMS estimation of process variables. Let $y \in \mathcal{R}^m, x \in \mathcal{R}^n$ be two random vectors satisfying

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right), \Sigma_{yx} = \Sigma_{xy}^T.$$

Then, by the CCA algorithm described in Sub-section 3.3.1, an LMS estimate for x using the process measurement vector y is given by

$$\hat{x} = \Sigma_{xy} L L^T y = \Sigma_{xy} \Sigma_y^{-1} y. \tag{3.98}$$

Moreover,

$$\hat{x}_N = J^T \Sigma_{xy} L L^T y = R^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1} y$$

delivers an estimation for the normalised process vector x,

$$x_N = J^T x \sim \mathcal{N}(0, I),$$

which is useful to build the test statistic for the fault detection purpose.

Recall that the core of PLS algorithms is the computation of the "mostly correlated" eigenvectors of the estimated Σ_{xy}. In this context, PLS and CCA are similar. Below, we introduce an alternative CCA algorithm, which delivers an LS estimate of x using collected data of y and is numerically reliable like PLS algorithms. To this end, we first introduce a known result.

Theorem 3.5 *Given*

$$\Sigma_x \in \mathcal{R}^{n \times n}, \Sigma_x > 0, \Sigma_y \in \mathcal{R}^{m \times m}, \Sigma_y > 0, \Sigma_{xy} = \Sigma_{yx}^T \in \mathcal{R}^{n \times m},$$

and assume that

$$rank\left(\Sigma_{xy}\right) = \kappa.$$

Then, the canonical correlation vectors, $J_i, L_i, i = 1, \cdots, \kappa$, defined in Definition 3.1, are the solution of the following optimisation problem

$$\max_{\bar{J},\bar{L}} tr\left(\bar{J}^T \Sigma_{xy} \bar{L}\right) \tag{3.99}$$

$$s.t.\ \bar{J}^T \Sigma_x \bar{J} = I_{\kappa\times\kappa}, \bar{L}^T \Sigma_y \bar{L} = I_{\kappa\times\kappa}, \tag{3.100}$$

where $\bar{J} \in \mathcal{R}^{n\times\kappa}, \bar{L} \in \mathcal{R}^{m\times\kappa}$.

The proof of this theorem is evident. In fact, due to constraint (3.100), $\bar{J}$, $\bar{L}$ can be parameterised as

$$\bar{J} = \Sigma_x^{-1/2} U_x, U_x^T U_x = I_{\kappa\times\kappa},$$
$$\bar{L} = \Sigma_y^{-1/2} U_y, U_y^T U_y = I_{\kappa\times\kappa}.$$

It yields

$$tr\left(\bar{J}^T \Sigma_{xy} \bar{L}\right) = tr\left(U_x^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} U_y\right).$$

Let

$$\bar{K} = \begin{bmatrix} \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} & 0 \\ 0 & 0 \end{bmatrix} \in \mathcal{R}^{\varsigma\times\varsigma}, \bar{U}_y = \begin{bmatrix} U_y \\ 0 \end{bmatrix} \in \mathcal{R}^{\varsigma\times\kappa},$$
$$\bar{U}_x^T = \begin{bmatrix} U_x^T & 0 \end{bmatrix} \in \mathcal{R}^{\kappa\times\varsigma}, \varsigma = \min(m, n) \geq \kappa.$$

Note that

$$U_x^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} U_y = \bar{U}_x^T \bar{K} \bar{U}_y \in \mathcal{R}^{\kappa\times\kappa},$$
$$\sigma_i\left(\bar{K}\right) = \sigma_i\left(K\right) = \sigma_i, i = 1, \cdots, \kappa,$$
$$\sigma_i\left(\bar{U}_y \bar{U}_x^T\right) = \sigma_i\left(U_y U_x^T\right) = 1, i = 1, \cdots, \kappa,$$

where K is defined in (3.22). It turns out

$$tr\left(U_x^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} U_y\right) = tr\left(\bar{U}_y \bar{U}_x^T \bar{K}\right)$$
$$\leq \sum_{i=1}^{\kappa} \sigma_i\left(\bar{U}_y \bar{U}_x^T\right) \sigma_i\left(\bar{K}\right) = \sum_{i=1}^{\kappa} \sigma_i\left(K\right).$$

On the other hand, for

$$U_x = \begin{bmatrix} r_1 & \cdots & r_\kappa \end{bmatrix}, U_y = \begin{bmatrix} v_1 & \cdots & v_\kappa \end{bmatrix},$$

with $r_i, v_i, i = 1, \cdots, \kappa$, being defined in Definition 3.1, it holds

$$tr\left(U_x^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} U_y\right) = tr\left(\Sigma\right) = \sum_{i=1}^{\kappa} \sigma_i\left(K\right).$$

That is,

$$\bar{J} = \begin{bmatrix} J_1 & \cdots & J_\kappa \end{bmatrix}, \bar{L} = \begin{bmatrix} L_1 & \cdots & L_\kappa \end{bmatrix}$$

solve the optimisation problem (3.99)–(3.100).

We now consider the estimation issue. Similar to the PLS study, we assume that $m >> n$ and $\kappa = n$. Note that

$$\bar{J}^T \Sigma_{xy} L = \bar{J}^T \Sigma_{xy} \begin{bmatrix} \bar{L} & \tilde{L} \end{bmatrix} = \begin{bmatrix} \bar{\Sigma} & 0 \end{bmatrix},$$

where

$$\tilde{L} = \begin{bmatrix} L_{n+1} & \cdots & L_m \end{bmatrix}.$$

It yields

$$\bar{J}^T \Sigma_{xy} \tilde{L} = 0 \Longrightarrow \bar{J}^T \Sigma_{xy} \bar{L} \bar{L}^T y = \bar{J}^T \Sigma_{xy} L L^T y = R^T \Sigma_x^{-1/2} \Sigma_{xy} \Sigma_y^{-1/2} y.$$

As a result, using $\bar{J}$, $\bar{L}$ delivered by solving the optimisation problem (3.99), we are able to construct an LMS estimation for the normalised process variable x, that is

$$\hat{x}_N = \bar{J}^T \Sigma_{xy} \bar{L} \bar{L}^T y. \tag{3.101}$$

The data-driven realisation for $\hat{x}_N$ is given in the following algorithm.

Algorithm 3.3 *Estimation of the normalised process variable x*

- *Data pre-processing for X_N, Y_N;*
- *Forming*

$$\Sigma_{xy} \approx \frac{1}{N-1} X_N Y_N^T, \, \Sigma_x \approx \frac{1}{N-1} X_N X_N^T, \, \Sigma_y \approx \frac{1}{N-1} Y_N Y_N^T;$$

- *Solving optimisation problem (3.99) for $\bar{J}$, $\bar{L}$;*
- *Estimating the normalised process variable x using (3.101).*

3.6 Notes and References

Although the fault detection and estimation issues addressed in this chapter are very basic, in which only static processes are under consideration and common statistic and linear algebraic methods are applied for the problem solutions, the ideas and concepts behind the solutions are fundamental. They provide us with a systematic way of handling optimal FD and FE problems, and can be applied for other types of systems, dynamic, time-varying or nonlinear systems, which will be investigated in the subsequent chapters.

To begin with, a basic problem of detecting and estimating faults in the mean of static processes, as formulated in Definition 2.4, and its general form have been studied. The well-known solution with χ^2 test statistic is achieved based on Neyman-Pearson Lemma, which proves, roughly speaking, the likelihood ratio delivers the best fault detectability, when the likelihood function of the fault is known. Neyman-Pearson Lemma is a well-known result in hypothesis testing methods. For more details about it, the reader is referred to [1]. In the application to fault detection, the likelihood function of the fault to be detected is in general partially known, this requires (i) an optimal fault estimation (for instance MLE), and (ii) embedding the fault estimate into the likelihood ratio. The achieved χ^2 test statistic is the result of this solution procedure. This is the basic idea and application procedure of the GLR method, for which we refer the reader to the excellent monographs by Basseville and Nikiforov [2], and by Gustafsson [3].

One point should be emphasised that a direct application of the estimated fault to building test statistic for the FD purpose will not, in general, lead to an optimal fault detection. On the other hand, the first step of performing MLE of f is of considerable importance for the FD performance, in particular when the general form of the process model (3.10) is under consideration. From the FD point of view, this step can be interpreted as mapping the measurement vector to a subspace with reduced influence of the noise without loss of the sensitivity to the fault.

In recent years, application of CCA technique to fault detection has become popular, thanks to the so-called residual forms (3.33) and (3.30) [4]. CCA is a standard MVA analysis method. As introduced in Sub-section 3.3.1, which is a summary of Chap. 15 in [5], CCA describes the correlation relation between two random vectors. In [4], it has been proposed to make use of the canonical correlation vectors to build the residual model and apply it for fault detection. The idea behind this FD scheme is the reduction of the uncertainty in sense of variance in the process measurement, which is well illustrated by (3.34)–(3.35). In fact, it is well-known that if two random variables are correlated, we can use one variable to estimate the other one. When the estimate is LMS, the covariance of the estimation error is minimum. As shown in (3.36)–(3.37), CCA delivers such an estimate. In this sense, the residual model (3.33) or (3.30) results in optimal FD. For the same reason and in this context, we have the equivalence between the CCA and GLR methods, as demonstrated in Sub-section 3.3.3.

It should be stressed with all clarity that the idea of applying MVA methods to dealing with FD issues is to extract information about uncertainty in the process measurement variables. In the context of FD studies, the uncertainty is expressed in terms of the covariance matrix and serves as a weighting matrix in the χ^2 test statistic to increase the sensitivity to the faults. In the CCA method, the covariance matrix is adopted to construct an estimate of a random (measurement) random variable using another (measurement) random variable aiming at reducing the uncertainty (again expressed in terms of its variance) in the estimated random variable.

In comparison with FD studies on processes with measurement noises, research work on FD in processes with deterministic disturbances has rarely been reported. One reason could be that there exists no established framework for such investigations. Under this consideration, our efforts have been devoted to solving the optimal FD problems formulated in Definitions 2.7 and 2.9, which serve as a frame for our study. The achieved solutions are given in Theorems 3.2–3.3. Some interesting aspects are summarised as follows:

- like the handling of statistic FD problems, an LS fault estimation builds the first step towards reducing the influence of the unknown disturbance,
- it is followed by switching on the right inverse of the mapping from the disturbance to the measurement,
- by using $\|\cdot\|^2$ as the evaluation function, the threshold is set to be the bound of the disturbance δ_d^2.

The computation of finding the right inverse of the mapping from the disturbance to the measurement is a key step of the solution for the optimal FD problem given in Definition 2.7. It can be understood as an LS estimate of the disturbance d, whose norm, as the LS solution, is bounded by δ_d. This enables us to make use of available information about the disturbance d to achieve optimal fault detection.

The so-called data-driven solutions of FD and FE problems addressed in Sect. 3.5 deal with the identical FD and FE problems presented in the previous sections on the same model assumptions, but without *a priori* knowledge of the model parameters. Instead, great number of process operation data are collected and recorded. It is worth mentioning that the model assumptions are essential for the test statistic building and threshold setting, although they are not explicitly mentioned in most of the publications on these issues. In such a case, it is intuitional to use the recorded data for the purpose of identifying the (unknown) model parameters, which then allows us to apply the existing (model-based) statistical methods to detect and estimate the faults. In general, data-driven FD schemes consist of two phases: (i) (offline) training aiming to identify the model parameters, and (ii) online detection and estimation. In application practice, it is the state of the art that the recorded process data are assumed to be collected during the fault-free operations. As a result, on the assumption of process model (3.1) or (3.10), the mean value and covariance matrix of the measurement noise are identified using the recorded process data.

In Sub-section 3.5.2, we have studied the data-driven fault detection problem, in which the "training data" and online measurement data are treated as two independent samples and the detection problem is formulated as detecting the difference (change) between the mean values of the samples. By means of the T^2 test statistic for change detection, the threshold setting is realised using $\mathcal{F}$-distribution. We refer the reader to [5–7] for the needed mathematical knowledge and detailed discussion in this subsection. The proof of Theorem 3.4 can be found, for instance, in Sect. 5.2 in [5]. As an alternative statistic, we have introduced Q test statistic in Subsection 3.5.3. This test statistic has been, for instance, proposed by Nomikos and MacGregor for process monitoring [8], where the T^2 test statistic is, considering the involved inverse computation of the covariance matrix, replaced by the Q test statistic for a reliable monitoring performance. For the purpose of threshold setting, the χ^2 approximations of the Q test statistic that was proposed and proved by Box [9] has been adopted.

PCA is a basic MVA method and widely applied to data compression, feature extraction, image processing, pattern recognition, signal analysis and process monitoring [10]. The application of the basic PCA algorithms, as summarised in Sub-section 3.5.4, to fault detection can be found, for example, in [11–13]. We have illustrated that the basic PCA algorithms are, for their FD application, a special case of χ^2 or T^2 test statistic. The SVD performed in the PCA offline computation algorithm serves as numerical computation of the inverse of the covariance matrix of the measurements. It does not lead to any improvement of fault detection performance.

In the past two decades, numerous variations of PCA methods have been reported. A recent development is the application of the PPCA technique [14] to process monitoring and fault detection [15]. We have briefly discussed about PPCA application in the context of detecting possible faults (changes in the mean value) using process data and on the assumption of model (3.92)–(3.93) without touching the modelling issue which is typically handled using the EM algorithm. This topic will also be addressed in Chap. 13. Further developments of the PCA technique include, for example, the recursive implementation of PCA [16], fast moving window PCA [17], kernel PCA [18], and dynamic PCA [12], just mentioning some representative algorithms.

PLS is a standard tool and widely used in chemometrics [19]. The first successful applications of PLS to fault diagnosis and process monitoring have been reported in [11, 20, 21]. Roughly speaking, PLS regression is an MVA method that constructs a linear regression between two random variables based on their correlation (expressed by the covariance matrix). It is evident that there exist close relations between the LS, CCA and PLS methods. This fact also motivates our work in Sub-section 3.5.5. In [22], it has been demonstrated that PLS performs an oblique projection instead of an orthogonal one, as done in the LS or CCA methods. Consequently, LS and CCA algorithms deliver an LMS estimate or minimum variance residual vector and thus perform an optimal FD, as illustrated in our study. Differently, the advantage of the standard PLS algorithm lies in the numerical reliability. There exist a number of variations of PLS regression algorithms [23–25], and remarkable efforts have been made recently to improve its application to process monitoring and fault diagnosis. In [26], recursive PLS (RPLS) has been proposed. Li et al. have studied the geometric properties of PLS for process monitoring [22]. Zhou et al. have developed the so-called T-PLS (total PLS) approach [27]. Benchmark and comparison studies on PLS and successful applications of PLS have been reported in [12, 28–30].

References

[1] E. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*. Springer, 2008.

[2] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes—Theory and Application*. New Jersey: Prentice-Hall, 1993.

[3] F. Gustafsson, *Adaptive Filtering and Change Detection*. Chichester: John Wiley and Sons, LTD, 2000.

[4] Z. Chen, S. X. Ding, K. Zhang, Z. Li, and Z. Hu, “Canonical correlation analysis-based fault detection methods with application to alumina evaporation process,” *Control Engineering Practice*, vol. 46, pp. 51–58, 2016.

[5] W. K. Härdle and L. Simar, *Applied Multivariate Statistical Analysis, Third Edition*. Berlin Heidelberg: Springer, 2012.

[6] H. M. Wadsworth, *Handbook of Statistical Methods for Engineers and Scientists, the 2nd Edition*. New York: McGraw-Hill, 1997.

[7] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Process, the Fourth Edition*. New York: McGraw-Hill, 2002.

[8] P. Nomikos and J. Macgregor, "Multivariate SPC charts for monitoring batch processes," *Technometrics*, vol. 37, pp. 41–59, 1995.

[9] G. Box, "Some theorems on quadratic forms applied in the study of analysis of variance problems: Effect of inequality of variance in one-way classification," *Annals of Mathematical Statistics*, vol. 25, pp. 290–302, 1954.

[10] I. Jolliffe, *Principal Component Analysis*. New York, Berlin: Springer-Verlag, 1986.

[11] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Contr. Eng. Practice*, vol. 3, pp. 403–414, 1995.

[12] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London: Springer, 2001.

[13] S. J. Qin, "Statistical process monitoring: Basics and beyond," *Journal of Chemometrics*, vol. 17, pp. 480–502, 2003.

[14] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. R. Statist. Soc. B*, vol. 61, pp. 611–622, 1999.

[15] D. Kim and I.-B. Lee, "Process monitoring based on probabilistic PCA," *Chemometrics and Intelligent Laboratory Systems*, vol. 67, pp. 109–123, 2003.

[16] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *Journal of Process Control*, vol. 10, pp. 471–486, 2000.

[17] X. Wang, U. Kruger, and G. W. Irwin, "Process monitoring approach using fast moving window PCA," *Ind. Eng. Chem. Res.*, vol. 44, pp. 5691–5702, 2005.

[18] X. Liu, U. Kruger, T. Lttler, L. Xie, and S. Wang, "Moving window kernel PCA for adaptive monitoring of nonlinear processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 96, pp. 132–143, 2009.

[19] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: A basic tool of chemometrics," *Chemometrics and Int. Lab. Syst.*, vol. 58, pp. 109–130, 2001.

[20] J. Kresta, J. MacGregor, and T. Marlin, "Multivariate statistical monitoring of process operating performance," *Canadian Journal of Chemical Engineering*, vol. 69, pp. 35–47, 1991.

[21] B. Wise and N. Gallagher, "The process chemometrics approach to process monitoring and fault detection," *Journal of Process Control*, vol. 6, pp. 329–348, 1996.

[22] G. Li, S. J. Qin, and D. Zhou, "Geometric properties of partial least squares for process monitoring," *Automatica*, 2010.

[23] I. S. Helland, "On the structure of partial least squares regression," *Communication in Statistics - Simulation and Computation*, vol. 17, pp. 581–607, 1988.

[24] A. Höskuldsson, "PLS regression methods," *Journal of Chemometrics*, vol. 2, pp. 211–228, 1988.

[25] B. Dayal and J. F. MacGregor, "Improved PLS algorithms," *Jounal of Chemometrics*, vol. 11, pp. 73–85, 1997.

[26] S. J. Qin, "Recursive PLS algorithms for adaptive data modeling," *Computers chem. Eng.*, vol. 22, pp. 503–514, 1998.

[27] D. Zhou, G. Li, and S. J. Qin, "Total projection to latent structures for process monitoring," *AIChE Journal*, vol. 56, pp. 168–178, 2010.

[28] G. Li, C. F. Alcala, S. J. Qin, and D. Zhou, "Generalized reconstruction-based contributions for output-relevant fault diagnosis with application to the Tennessee Eastman process," *IEEE Trans. on Contr. Syst. Tech.*, vol. 19, pp. 1114–1127, 2011.

[29] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, pp. 1567–1581, 2012.

[30] K. X. Peng, K. Zhang, G. Li, and D. Zhou, "Contribution rate plot for nonlinear quality-related fault diagnosis with application to the hot strip mill process," *Contr. Eng. Practice*, vol. 21, pp. 360–369, 2013.

Chapter 4
Basic Methods for Fault Detection in Dynamic Processes

Issues related to fault diagnosis in dynamic processes are important topics in the application and research domains. Thanks to their intimate relations to automatic control systems, model-based schemes are widely accepted as a powerful technology in dealing with process monitoring and fault diagnosis in dynamic processes. Triggered by the recent trend with machine learning and big data, data-driven design of model- and observer-based fault detection systems for dynamic processes draws remarkable research attention, in which various schemes have been proposed.

In this chapter, we first review some basic model-based methods for residual generation. It is followed by the introduction of two basic model-based methods for fault detection in dynamic processes. Our focus is on

- outlining the basic ideas and principles of model- and observer-based fault detection, which are of considerable importance for our subsequent study, and
- demonstrating how they solve the fault detection problems formulated in Chap. 2.

We will also present a data-driven design scheme for model- and observer-based fault detection systems.

4.1 Preliminaries and Review of Model-Based Residual Generation Schemes

Residual generation is an essential step for detecting faults in dynamic processes driven by certain input variables. Roughly speaking, residual generation is

- to build software redundancy for process measurement variables and
- to create a residual vector by comparing the process measurement variables and their software redundancy.

In the ideal case, the residual vector should contain information about all possible process variations caused by faults and disturbances, and is independent of the process input variables. The mostly popular way to build software redundancy is the use of

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_4

a nominal process model and, based on it, the construction of an observer, or in general, a residual generator.

4.1.1 Nominal System Models

A dynamic system can be described in different ways. The so-called linear time invariant (LTI) system model offers the simplest form and thus is widely used in research and application domains. We call disturbance-free and fault-free systems nominal, and suppose that the nominal systems are LTI. There are two basic mathematical model forms for LTI systems: the transfer function matrix and the state space representation.

Transfer function and state space representation

Roughly speaking, a transfer (function) matrix is an input-output description of the dynamic behaviour of an LTI system in the frequency domain. Throughout this book, notation $G_{yu}(z)$ is used for presenting a transfer matrix from the input vector $u \in \mathcal{R}^l$ to the output vector $y \in \mathcal{R}^m$, that is,

$$y(z) = G_{yu}(z)u(z). \tag{4.1}$$

It is assumed that $G_{yu}(z)$ is a proper real-rational matrix. We use z to denote the complex variable of z-transform for discrete-time signals.

The standard form of the state space representation of a discrete-time LTI system is

$$x(k+1) = Ax(k) + Bu(k), x(0) = x_0, \tag{4.2}$$
$$y(k) = Cx(k) + Du(k), \tag{4.3}$$

where $x \in \mathcal{R}^n$ is called the state vector, x_0 the initial condition of the system, $u \in \mathcal{R}^l$ the input vector and $y \in \mathcal{R}^m$ the output vector. Matrices A, B, C, D are appropriately dimensioned real constant matrices.

State space models can be either directly achieved by modelling or derived based on a transfer matrix. The latter is called a state space realisation of

$$G_{yu}(z) = C(zI - A)^{-1}B + D,$$

and denoted by

$$G_{yu}(z) = (A, B, C, D) \text{ or } G_{yu}(z) = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{4.4}$$

In general, we assume that (A, B, C, D) is a minimal realisation of $G_{yu}(z)$.

Remark 4.1 *Throughout the book, for the sake of simplicity, we may drop out of the (frequency or time) domain variable of a transfer function or a system variable, when this does not cause any confusion. For example, we may use G_{yu} instead of $G_{yu}(z)$, y instead of $y(k)$.*

Coprime factorisation technique

Coprime factorisation of a transfer function matrix gives a further system representation form which will be intensively used in our subsequent study. In simple words, a coprime factorisation over $\mathcal{RH}_\infty$ is to factorise a transfer matrix into two stable and coprime transfer matrices.

Definition 4.1 *Two stable transfer matrices $\hat{M}(z)$, $\hat{N}(z)$ are called left coprime if there exist two stable transfer matrices $\hat{X}(z)$ and $\hat{Y}(z)$ such that*

$$\left[\hat{M}(z)\ \hat{N}(z)\right]\begin{bmatrix}\hat{X}(z)\\ \hat{Y}(z)\end{bmatrix} = I. \tag{4.5}$$

Similarly, two stable transfer matrices $M(z)$, $N(z)$ are right coprime if there exist two stable matrices $Y(z)$, $X(z)$ such that

$$\left[X(z)\ Y(z)\right]\begin{bmatrix}M(z)\\ N(z)\end{bmatrix} = I. \tag{4.6}$$

Let $G(z)$ be a proper real-rational transfer matrix. The left coprime factorisation (LCF) of $G(z)$ is a factorisation of $G(z)$ into two stable and coprime matrices which will play a key role in designing the so-called residual generator. To complete the notation, we have also introduced the right coprime factorisation (RCF), which is however only occasionally applied in dealing with residual generation issues and mainly in the control relevant context.

Definition 4.2 *$G(z) = \hat{M}^{-1}(z)\hat{N}(z)$ with the left coprime pair $\left(\hat{M}(z), \hat{N}(z)\right)$ is called LCF of $G(z)$. Its dual form, RCF of $G(z)$, is defined by $G(z) = N(z)M^{-1}(z)$ with the right coprime pair $(M(z), N(z))$.*

Below are the computation formulas for $\left(\hat{M}(z), \hat{N}(z)\right)$, $(M(z), N(z))$ and the associated pairs $\left(\hat{X}(z), \hat{Y}(z)\right)$ and $(X(z), Y(z))$.

Suppose $G(z)$ is a proper real-rational transfer matrix with a state space realisation (A, B, C, D), and it is stabilisable and detectable. Let F and L be so that $A + BF$ and $A - LC$ are Schur matrices (that is, their eigenvalues are inside the unit circle on the complex plane). Then

$$\hat{M}(z) = (A - LC, -L, C, I), \hat{N}(z) = (A - LC, B - LD, C, D), \quad (4.7)$$
$$M(z) = (A + BF, B, F, I), N(z) = (A + BF, B, C + DF, D), \quad (4.8)$$
$$\hat{X}(z) = (A + BF, L, C + DF, I), \hat{Y}(z) = (A + BF, -L, F, 0), \quad (4.9)$$
$$X(z) = (A - LC, -(B - LD), F, I), Y(z) = (A - LC, -L, F, 0) \quad (4.10)$$

give the LCF and RCF of $G(z)$ as well as two other coprime pairs $\left(\hat{X}(z), \hat{Y}(z)\right)$ and $(X(z), Y(z))$, respectively. These eight transfer matrices build the so-called Bezout identity

$$\begin{bmatrix} X(z) & Y(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix} \begin{bmatrix} M(z) & -\hat{Y}(z) \\ N(z) & \hat{X}(z) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}. \quad (4.11)$$

4.1.2 Observer-Based Residual Generation Schemes

Next, we introduce two standard observer-based residual generation schemes.

Fault detection filter

Fault detection filter (FDF) is a type of observer-based residual generators proposed by Beard and Jones in the early 1970s. Their work marked the beginning of a stormy development of model-based FDI techniques.

The core of an FDF is a full-order state observer

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L\left(y(k) - C\hat{x}(k) - Du(k)\right), \quad (4.12)$$

which is constructed on the basis of the nominal system model

$$G_{yu}(z) = C(zI - A)^{-1}B + D.$$

Built upon (4.12), the residual is simply defined by

$$r(k) = y(k) - \hat{y}(k) = y(k) - C\hat{x}(k) - Du(k). \quad (4.13)$$

The advantages of an FDF lie in its simple construction form and, for the reader who is familiar with the modern control theory, in its intimate relationship with the state observer design and especially with the well-established robust control theory by designing robust residual generators.

We see that the design of an FDF is in fact the determination of the observer gain matrix L. To increase the degree of design freedom, we can switch a matrix to the output estimation error $y(z) - \hat{y}(z)$, that is

$$r(z) = V\left(y(z) - \hat{y}(z)\right). \tag{4.14}$$

Diagnostic observer scheme

The diagnostic observer (DO) is, thanks to its flexible structure and similarity to the Luenberger type observer, one of the mostly investigated model-based residual generator forms.

The core of a DO is a Luenberger type (output) observer described by

$$\xi(k+1) = G\xi(k) + Hu(k) + Ly(k), \tag{4.15}$$

$$r(k) = Vy(k) - W\xi(k) - Qu(k), \tag{4.16}$$

where $\xi \in \mathcal{R}^s$, s denotes the observer order and can be equal to or lower or higher than the system order n. Although most contributions to the Luenberger type observer are focused on the case with lower order aiming at getting a reduced order observer, higher order observers may play an important role in the optimisation of fault detection systems.

Assume $G_{yu}(z) = C(zI - A)^{-1}B + D$, then matrices G, H, L, Q, V and W together with a matrix $T \in \mathcal{R}^{s \times n}$ have to satisfy the so-called Luenberger conditions,

$$I.\ G \text{ is a Schur matrix}, \tag{4.17}$$

$$II.\ TA - GT = LC, H = TB - LD, \tag{4.18}$$

$$III.\ VC - WT = 0, Q = VD, \tag{4.19}$$

under which system (4.15)–(4.16) delivers a residual vector satisfying

$$\forall u, x(0), \lim_{k \to \infty} r(k) = 0.$$

Let

$$e(k) = T\xi(k) - x(k).$$

It is straightforward to show that the system dynamics of DO is governed by

$$e(k+1) = Ge(k), r(k) = Ve(k).$$

4.1.3 Parity Space Approach

The parity space approach was initiated by Chow and Willsky in their pioneering work in the early 1980's. The parity space approach is generally recognised as one of the important model-based residual generation approaches.

We consider in the following state space model (4.2)–(4.3) and, without loss of generality, assume

$$rank(C) = m.$$

Introducing the notations

$$y_s(k) = \begin{bmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k) \end{bmatrix}, u_s(k) = \begin{bmatrix} u(k-s) \\ u(k-s+1) \\ \vdots \\ u(k) \end{bmatrix}, \tag{4.20}$$

$$H_{o,s} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix}, H_{u,s} = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{s-1}B & \cdots & CB & D \end{bmatrix}, \tag{4.21}$$

a straightforward computation using the state space model (4.2)–(4.3) yields the following compact model form

$$y_s(k) = H_{o,s}x(k-s) + H_{u,s}u_s(k). \tag{4.22}$$

Note that (4.22) describes the input and output relationship in dependence on the past state vector $x(k-s)$, which is unknown (not measured). The underlying idea of the parity relation based residual generation lies in the utilisation of the fact, known from the linear control theory, that for $s \geq n$ the following rank condition holds:

$$rank\left(H_{o,s}\right) \leq n < \text{ the row number of matrix } H_{o,s} = (s+1)m.$$

This ensures that for $s \geq n$ there exists at least a (row) vector

$$v_s(\neq 0) \in \mathcal{R}^{(s+1)m}$$

such that

$$v_s H_{o,s} = 0. \tag{4.23}$$

Hence, a parity relation based residual generator is constructed by

$$r(k) = v_s\left(y_s(k) - H_{u,s}u_s(k)\right), \tag{4.24}$$

whose dynamics is governed by

$$r(k) = v_s\left(y_s(k) - H_{u,s}u_s(k)\right) = v_s H_{o,s}x(k-s) = 0.$$

Vectors satisfying (4.23) are called parity vectors, the set of which,

$$P_s = \{v_s \mid v_s H_{o,s} = 0\}, \tag{4.25}$$

is called the parity space of the s-th order.

One of the significant properties of parity relation based residual generators, also widely viewed as the main advantage over the observer-based approaches, is that the design can be carried out in a straightforward manner. In fact, it only deals with solutions of linear equations or linear optimisation problems. In against, the implementation form (4.24) is surely not optimal for an online realisation, since for the online computation not only the temporal but also a number of past measurement and input data are needed, which have to be recorded. In Sub-section 4.4.5, a one-to-one mapping between the parity space approach and the observer-based approach will be presented, which allows an observer-based residual generator construction for a given a parity vector. Based on this result, a strategy called *parity space design, observer-based implementation* has been developed, which makes use of the computational advantage of parity space approaches for the system design (selection of a parity vector or matrix) and then realises the solution in the observer form to ensure a numerically stable and less consuming on-line computation. This strategy has been, for instance, successfully used in the sensor fault detection in vehicles and highly evaluated by engineers in industry. It is worth mentioning that the strategy of *parity space design, observer-based implementation* can also be applied to continuous-time systems.

4.1.4 Kernel Representation and Parameterisation of Residual Generators

In the model-based fault detection framework, the FDF and DO based residual generators are called closed-loop configurated, since a feedback of the residual signal is embedded in the system configuration and the computation is realised in a recursive form. Differently, the parity space residual generator is of an open-loop structure. In fact, it is an FIR (finite impulse response) filter. Below, we briefly introduce a general form for all types of LTI residual generators, which is also called parameterisation of residual generators.

A fundamental property of the LCF is that in the fault- and disturbance-free case

$$\forall u, \begin{bmatrix} -\hat{N}(z) & \hat{M}(z) \end{bmatrix} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = 0. \tag{4.26}$$

Equation (4.26) is called kernel representation (KR) of the system under consideration and useful in parameterising all residual generators. For our purpose, we introduce below a more general definition of kernel representation.

Definition 4.3 *Given system (4.2)–(4.3), a stable linear system $\mathcal{K}$ driven by $u(z)$, $y(z)$ and satisfying*

$$\forall u(z), r(z) = \mathcal{K} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = 0 \tag{4.27}$$

is called stable kernel representation (SKR) of system (4.2)–(4.3).

It is clear that system $\left[-\hat{N}(z) \ \hat{M}(z) \right]$ is an SKR. Now, consider the process model (4.2)–(4.3). A parameterisation form of all LTI residual generators is described by

$$r(z) = R(z) \left[-\hat{N}(z) \ \hat{M}(z) \right] \begin{bmatrix} u(z) \\ y(z) \end{bmatrix}, \tag{4.28}$$

where $R(z)$ $(\neq 0)$ is a stable parameterisation system and called post-filter. Moreover, in order to avoid loss of information about faults to be detected, in general, the condition

$$rank(R(z)) = m$$

is to be satisfied.

It has been demonstrated that

$$\left[-\hat{N}(z) \ \hat{M}(z) \right] \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = y(z) - \hat{y}(z) \tag{4.29}$$

with $\hat{y}$ being delivered by a full order observer as an estimate of y. Consequently, we can apply an FDF,

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L\left(y(k) - \hat{y}(k)\right), \hat{y}(k) = C\hat{x}(k) + Du(k),$$

for the realisation of (4.28). Based on this result, it is also straightforward to prove the following relation between the observer setting and post-filter.

Lemma 4.1 *Given*

$$\begin{gathered} r_1(z) = \hat{M}_1(z)y(z) - \hat{N}_1(z)u(z), r_2(z) = \hat{M}_2(z)y(z) - \hat{N}_2(z)u(z) \\ \hat{M}_i(z) = \left(A_{L_i}, -L_i, C, I\right), \hat{N}_i(z) = \left(A_{L_i}, B - L_iD, C, D\right) \\ A_{L_i} = A - L_iC, i = 1, 2, \end{gathered}$$

then, it holds

$$\begin{gathered} \left[\hat{M}_2(z) \ \hat{N}_2(z) \right] = Q_{21}(z) \left[\hat{M}_1(z) \ \hat{N}_1(z) \right] \Longrightarrow r_2(z) = Q_{21}(z) r_1(z), \\ Q_{21}(z) = I - C\left(zI - A_{L_2}\right)^{-1}(L_2 - L_1). \end{gathered} \tag{4.30}$$

Moreover, $Q_{21}(z)$ is invertible, and $Q_{21}^{-1}(z)$ is stable and satisfies

$$Q_{21}^{-1}(z) = I - C\left(zI - A_{L_1}\right)^{-1}(L_1 - L_2) = Q_{12}(z),$$

which gives

$$\left[\hat{M}_1(z)\ \hat{N}_1(z)\right] = Q_{12}(z)\left[\hat{M}_2(z)\ \hat{N}_2(z)\right] \Longrightarrow r_1(z) = Q_{12}(z)r_2(z).$$

Remark 4.2 *Consider a parity relation based residual vector $r(k)$,*

$$r(k) = V_s\left(y_s(k) - H_{u,s}u_s(k)\right),$$

where $V_s \in \mathcal{R}^{m\times(s+1)m}$ is the parity matrix. Let

$$V_s = \left[V_{s,0} \cdots V_{s,s}\right], V_{s,i} \in \mathcal{R}^{m\times m}, i = 0, 1, \cdots, s,$$

and assume

$$rank\left(V_{s,s}\right) = m.$$

Then, $r(k)$ can also be parameterised in the form

$$r(k) = V_s\left(y_s(k) - H_{u,s}u_s(k)\right) = V\left(y(k) - \hat{y}(k)\right),$$

by defining

$$\begin{aligned} V &= V_{s,s}, \hat{y}(k) = V_{s,s}^{-1}\left(V_s H_{u,s}u_s(k) - V_{s-1}y_{s-1}(k-1)\right), \\ V_{s-1} &= \left[V_{s,0} \cdots V_{s,s-1}\right], y_{s-1}(k-1) = \begin{bmatrix} y(k-s) \\ \vdots \\ y(k-1) \end{bmatrix}. \end{aligned}$$

4.2 Fault Detection in Linear Stochastic Processes

Consider the process model

$$x(k+1) = A(k)x(k) + B(k)u(k) + E(k)w(k), \tag{4.31}$$

$$y(k) = C(k)x(k) + D(k)u(k) + v(k), \tag{4.32}$$

where $x(k) \in \mathcal{R}^n, u(k) \in \mathcal{R}^p, y(k) \in \mathcal{R}^m$ are process state, input and output vectors, respectively, and all system matrices are of appropriate dimensions and known. $w(k) \in \mathcal{R}^{k_w}$, $v(k)$ are process and measurement noise vectors. It is assumed that they are uncorrelated with the state and input vectors, and

$$w(k) \sim \mathcal{N}\left(0, \Sigma_w(k)\right), v(k) \sim \mathcal{N}\left(0, \Sigma_v(k)\right),$$

$$\mathcal{E}\left(\begin{bmatrix} w(i) \\ v(i) \\ x(0) \end{bmatrix}\begin{bmatrix} w(j) \\ v(j) \\ x(0) \end{bmatrix}^T\right) = \begin{bmatrix} \begin{bmatrix} \Sigma_w(i) & S_{wv}(i) \\ S_{wv}^T(i) & \Sigma_v(i) \end{bmatrix}\delta_{ij} & 0 \\ 0 & \Pi_0 \end{bmatrix}, \delta_{ij} = \begin{cases} 1, i = j, \\ 0, i \neq j. \end{cases}$$

It is well-known that a Kalman filter with the recursive algorithm,

$$\begin{gathered} \hat{x}(k+1\left|k\right.) = A(k)\hat{x}(k\left|k-1\right.) + B(k)u(k) + K(k)r(k), \\ r(k) = y(k) - \hat{y}(k\left|k-1\right.), \hat{x}(0) = 0, \\ \hat{y}(k\left|k-1\right.) = C(k)\hat{x}(k\left|k-1\right.) + D(k)u(k), \\ K(k) = \left(A(k)P(k\left|k-1\right.)C^T(k) + E(k)S_{wv}(k)\right)\Sigma_r^{-1}(k), \\ P(k+1\left|k\right.) = A(k)P(k\left|k-1\right.)A^T(k) + E(k)\Sigma_w(k)E^T(k) \\ -K(k)\Sigma_r(k)K^T(k), \\ \Sigma_r(k) = C(k)P(k\left|k-1\right.)C^T(k) + \Sigma_v(k) = \mathcal{E}\left(r(k)r^T(k)\right), \end{gathered}$$

delivers a residual vector $r(k) \in \mathcal{R}^m$, which is white and of minimum covariance matrix.

The whiteness of the residual vector allows us to approach the fault detection problem at each time instant equivalently to the basic fault detection problem described in Sect. 3.1 with the model

$$r(k) = f(k) + \varepsilon(k), \varepsilon(k) \sim \mathcal{N}(0, \Sigma_r(k)), \tag{4.33}$$

where $f(k)$ represents any type of possible faults in the process or in the sensors and actuators. As described in Sect. 3.1, for given model (4.33) and an acceptable FAR α, setting $\{J, J_{th}\}$ equal to

$$J(k) = r^T(k)\Sigma_r^{-1}(k)r(k), J_{th} = \chi_\alpha^2, \tag{4.34}$$

leads to the optimal solution for *fault detection with maximum fault detectability,* as defined in (2.5). It is remarkable that the property with the minimum covariance matrix $\Sigma_r(k)$ results in overall maximal fault detectability. That is, given system model (4.31)–(4.32), the above Kalman filter based fault detection system delivers, in comparison with all other potential (linear) fault detection systems, the best fault detectability at the FAR level α.

4.3 Fault Detection in Linear Processes with Unknown Inputs

Consider the process model

$$x(k+1) = Ax(k) + Bu(k) + E_f f(k) + E_d d(k), \tag{4.35}$$

$$y(k) = Cx(k) + Du(k) + F_f f(k) + F_d d(k), \tag{4.36}$$

where $x(k)$, $u(k)$, $y(k)$ are as given in model (4.31)–(4.32), $d(k) \in \mathcal{R}^{k_d}$, $f(k) \in \mathcal{R}^{k_f}$ represent unknown input and fault vectors, respectively. It is assumed that $d(k)$ is l_2 bounded with

$$\|d\|_2^2 \leq \delta_d^2. \tag{4.37}$$

Fault detection issues for processes modelled by (4.35)–(4.36) have been extensively investigated. Our objectives in this section are

- to introduce the so-called "unified solution",
- to demonstrate that this unified solution solves the optimal fault detection problem given in Definition 2.7,
- to present an interpretation of the unified solution from an alternative viewpoint, and finally
- to give a dual form of the unified solution, which solves the optimal fault detection problem given in Definition 2.9.

4.3.1 A Basic Form of the Unified Solution

Consider the process model (4.35)–(4.36). Under the assumption

$$rank\left(G_{yf}(z)\right) = m, G_{yf}(z) = C\left(zI - A\right)^{-1} E_f + F_f, \tag{4.38}$$

$$\forall \theta \in [0, 2\pi], rank \begin{bmatrix} A - e^{j\theta} I & E_d \\ C & F_d \end{bmatrix} = n + m, \tag{4.39}$$

it is proved that the residual generator

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L_2\left(y(k) - \hat{y}(k)\right), \tag{4.40}$$

$$r(k) = V_r\left(y(k) - \hat{y}(k)\right), \hat{y}(k) = C\hat{x}(k) + Du(k), \tag{4.41}$$

$$L_2 = \left(AXC^T + E_d F_d^T\right) V_r^2, V_r = \left(CXC^T + F_d F_d^T\right)^{-1/2}, \tag{4.42}$$

where $X > 0$ solves the Riccati equation

$$AXA^T - X + E_d E_d^T - L_2\left(CXC^T + F_d F_d^T\right) L_2^T = 0, \tag{4.43}$$

is optimal in the sense of

$$\forall\theta \in [0, 2\pi], \{L_2, V_r\} = \arg\sup_{L,V} \frac{\sigma_i\left(V\hat{N}_f(e^{j\theta})\right)}{\left\|V\hat{N}_d\right\|_\infty}, i = 1, \cdots, m, \tag{4.44}$$

$$\hat{N}_f(z) = F_f + C\left(zI - A + LC\right)^{-1}\left(E_f - LF_f\right),$$
$$\hat{N}_d(z) = F_d + C\left(zI - A + LC\right)^{-1}\left(E_d - LF_d\right)$$

with $\sigma_i\left(V\hat{N}_f(e^{j\theta})\right)$ denoting the singular values of $V\hat{N}_f(e^{j\theta})$. $V\hat{N}_f(z)$, $V\hat{N}_d(z)$ represent the transfer functions from f and d to r, respectively. Since this solution is a unified and generalised form for the so-called $\mathcal{H}_-/\mathcal{H}_\infty$ and $\mathcal{H}_\infty/\mathcal{H}_\infty$ solutions, it is called unified solution. The optimisation problem formulated in (4.44) is also called $\mathcal{H}_i/\mathcal{H}_\infty$ optimisation.

Moreover, it is known that $V_r\hat{N}_d$ satisfies

$$\forall\theta \in [0, 2\pi], V_r\hat{N}_d(e^{j\theta})\left(V_r\hat{N}_d(e^{-j\theta})\right)^T = I, \tag{4.45}$$

which is called co-inner. As a result, in the fault-free case,

$$\forall d,\ \ \|r\|_2^2 = \left\|V_r\hat{N}_d\right\|_2^2 \leq \|d\|_2^2. \tag{4.46}$$

This suggests the residual evaluation function

$$J = \|r\|_2^2 \tag{4.47}$$

and threshold

$$J_{th} = \delta_d^2. \tag{4.48}$$

4.3.2 Optimality of the Unified Solution

We now demonstrate that the unified solution solves the optimal fault detection problem given in Definition 2.7. To this end, Theorem 2.1 is used. Recall that for the process model (4.35)–(4.36), the image of the disturbance vector and the fault domain, as defined in (2.12) and Definition 2.7, are

$$\mathcal{I}_d = \left\{y_d \left| y_d = G_{yd}(z)d(z), \forall d \in \mathcal{D}_d, \|d\|_2 \leq \delta_d \right.\right\},$$
$$\mathcal{D}_{f,undetc} = \left\{f \left| y_f = G_{yf}(z)f(z) \in \mathcal{I}_d \right.\right\},$$

where

$$G_{yd} = F_d + C\left(zI - A\right)^{-1} E_d.$$

That means in turn

$$\mathcal{M}_d = G_{yd}(z), \mathcal{M}_f = G_{yf}(z).$$

On the other hand, it is well-known that the dynamics of the residual generator is governed by

$$r(z) = V\hat{N}_f(z)f(z) + V\hat{N}_d(z)d(z), \tag{4.49}$$

$$V\hat{M}(z)G_{yf}(z) = V\hat{N}_f(z), V\hat{M}(z)G_{yd}(z) = V\hat{N}_d(z) \tag{4.50}$$

with

$$\hat{M}(z) = I - C\left(zI - A + LC\right)^{-1} L. \tag{4.51}$$

We now prove that for

$$L = L_2, V = V_r,$$

as given in (4.42),

$$\mathcal{M}_d^- = V_r\hat{M}(z)$$

satisfies the three conditions given in Theorem 2.1. It is evident that $\mathcal{M}_d^-$ is invertible and, as given in (4.46),

$$\forall d, \ \ \|r\|_2^2 = \left\|\mathcal{M}_d^- \circ \mathcal{M}_d(d)\right\|_2^2 = \left\|V_r\hat{N}_d\right\|_2^2 \leq \|d\|_2^2 .$$

Thus, Conditions (i) and (ii) are satisfied. Next, considering that $V_r\hat{N}_d(z)$ is co-inner, it holds

$$\forall r, \exists d \text{ s.t. } \ \|r\|_2^2 = \left\|V_r\hat{N}_d d\right\|_2^2 = \|d\|_2^2 .$$

Therefore, $\forall f$ leading to

$$\left\|r_f\right\|_2^2 = \left\|V_r\hat{N}_f f\right\|_2^2 \leq \delta_d^2,$$
$$r_f = V_r\hat{M}(e^{j\theta})G_{yf}(e^{j\theta})f(e^{j\theta}) = V_r\hat{N}_f(e^{j\theta})f(e^{j\theta}),$$

$\exists d$ so that

$$\left\|r_f\right\|_2^2 = \left\|V_r\hat{N}_f f\right\|_2^2 = \left\|V_r\hat{N}_d d\right\|_2^2 .$$

That means, Condition (iii) with (2.22) is also satisfied. As a result, we have proved the following theorem.

Theorem 4.1 *Given the process model (4.35)–(4.36) and the residual generator (4.40)–(4.42), then* $\{J, J_{th}\}$ *given by*

$$J = \|r\|_2^2, J_{th} = \delta_d^2 \tag{4.52}$$

deliver the optimal solution for the fault detection problem given in Definition 2.7.

In a similar way, we can also find an optimal solution to the dual fault detection problem given in Definition 2.9, which is summarised in the following corollary.

Corollary 4.1 *Given the process model (4.35)–(4.36), the margin of detectable faults β and the FDF (4.12)–(4.14), and assume that*

$$G_{yf}(z) \in \mathcal{C}^{m \times k_f}, rank\left(G_{yf}(z)\right) = m, \tag{4.53}$$

$$\forall \theta \in [0, 2\pi], rank \begin{bmatrix} A - e^{j\theta} I & E_f \\ C & F_f \end{bmatrix} = n + m. \tag{4.54}$$

Then, $\{L, V, J, J_{th}\}$ given by

$$L = \left(AXC^T + E_f F_f^T\right) V^2, V = \left(CXC^T + F_f F_f^T\right)^{-1/2}, \tag{4.55}$$

$$J = \|r\|_2^2, J_{th} = \beta^2 \tag{4.56}$$

solve the optimal fault detection problem given in Definition 2.9, where $X > 0$ solves the Riccati equation

$$AXA^T - X + E_f E_f^T - L\left(CXC^T + F_f F_f^T\right) L^T = 0. \tag{4.57}$$

Proof It is clear that the dynamics of the residual generator (FDF) is governed by

$$r(z) = V\hat{N}_f(z) f(z) + V\hat{N}_d(z) d(z) \tag{4.58}$$

where $\hat{N}_f(z), \hat{N}_d(z)$ are as given in (4.44). Moreover, the FDF gain matrix and the post-filter $\{L, V\}$ given in (4.55) result in, analogous to the unified solution, that $V\hat{N}_f(z)$ is co-inner. That is

$$\forall \theta \in [0, 2\pi], V\hat{N}_f(e^{j\theta}) \left(V\hat{N}_f(e^{-j\theta})\right)^T = I. \tag{4.59}$$

Hence,

$$\forall f, \left\|r_f\right\|_2^2 = \left\|V\hat{N}_f f\right\|_2^2 = \|f\|_2^2. \tag{4.60}$$

Recall that

$$\mathcal{M}_f = G_{yf}(z), V\hat{M}(z) G_{yf}(z) = V\hat{N}_f(z).$$

Thus, $V\hat{M}(z)$ is $\mathcal{M}_f^-$ that yields

$$r_f = \mathcal{M}_f^- \mathcal{M}_f(f)$$

satisfying (4.60). Finally, according to Corollary 2.1, it is proved that $\{L, V\}$ and $\{J, J_{th}\}$ given by (4.55)–(4.56) solve the optimal fault detection problem given in Definition 2.9.

4.3.3 An Alternative Interpretation and Solution Scheme

Analogue to the solutions for the optimal fault detection in static processes, as discussed in Sect. 3.4, the solution to the fault detection problem given in Definition 2.7 can be interpreted as finding a generalised inverse of G_{yd} or an optimal estimation of d . To this end, the co-inner-outer factorisation serves as a useful tool. By a co-inner-outer factorisation of G_{yd},

$$G_{yd}(z) = \left(V_r\hat{M}(z)\right)^{-1} V_r\hat{N}_d(z)$$

with $V_r\hat{N}_d$ as co-inner and $\left(V_r\hat{M}(z)\right)^{-1}$ co-outer, $V_r\hat{M}(z)$ can be interpreted as a generalised inverse of $G_{yd}(z)$. Correspondingly, the evaluation function, in the fault-free case,

$$\|r\|_2^2 = \left\|V_r\hat{M}d\right\|_2^2$$

is a generalised form of (3.56).

Similarly, we can also view $\|r\|_2^2$ as an estimate for $\|d\|_2^2$. The relation (4.46) allows us to use available information about the disturbance, namely, its l_2 norm boundedness δ_d, for the threshold setting. That is

$$J_{th} = \delta_d^2.$$

In our subsequent investigation on optimal solutions for detecting faults in various types of dynamic systems, for instance, for time-varying systems and a class of nonlinear systems, we shall derive the solutions directly applying co-inner-outer factorisation technique.

4.4 A Data-Driven Method for Fault Detection in Dynamic Processes

We now review a data-driven method for detecting faults in dynamic processes. The core of this approach is the identification of system kernel representation, which has been, in the initial form, called subspace technique aided identification of parity vectors.

4.4.1 An Input-Output Data Model

For our purpose, we first introduce an input-output (I/O) data model. It is essential in our subsequent study and builds a link between the model-based fault detection schemes introduced in the previous sections and the data-driven design method to be introduced below. For our purpose, the following LTI process model is assumed to be the underlying model form adopted in our study

$$x(k+1) = Ax(k) + Bu(k) + w(k), \tag{4.61}$$

$$y(k) = Cx(k) + Du(k) + v(k), \tag{4.62}$$

where $u \in \mathcal{R}^p$, $y \in \mathcal{R}^m$ and $x \in \mathcal{R}^n$, $w \in \mathcal{R}^n$ and $v \in \mathcal{R}^m$ denote white noise sequences that are statistically independent of u and $x(0)$.

Let $\omega(k) \in \mathcal{R}^\xi$ be a data vector. We introduce the following notations:

$$\omega_s(k) = \begin{bmatrix} \omega(k-s) \\ \vdots \\ \omega(k) \end{bmatrix} \in \mathcal{R}^{(s+1)\xi}, \Omega_k = \begin{bmatrix} \omega(k) & \cdots & \omega(k+N-1) \end{bmatrix} \in \mathcal{R}^{\xi \times N}, \tag{4.63}$$

$$\Omega_{k,s} = \begin{bmatrix} \omega_s(k) & \cdots & \omega_s(k+N-1) \end{bmatrix} = \begin{bmatrix} \Omega_{k-s} \\ \vdots \\ \Omega_k \end{bmatrix} \in \mathcal{R}^{(s+1)\xi \times N}, \tag{4.64}$$

where s, N are some (large) integers. Note that by the notation $\omega_s(k)$ the time instant k in the bracket denotes the end time instant of the ω vector, and sub-index s is used in denoting the start time instant $k-s$. In our study, $\omega(k)$ can be $y(k)$, $u(k)$, $x(k)$, $w(k)$, $v(k)$, and ξ represents m or p or n given in (4.61)–(4.62).

By a straightforward computation using the state space model (4.61)–(4.62) and the notations (4.63)–(4.64), we have a data model of the form

$$Y_{k,s} = \Gamma_s X_{k-s} + H_{u,s} U_{k,s} + H_{w,s} W_{k,s} + V_{k,s} \in \mathcal{R}^{(s+1)m \times N}, \tag{4.65}$$

$$X_{k-s} = \begin{bmatrix} x(k-s) & \cdots & x(k-s+N-1) \end{bmatrix}, \Gamma_s = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix} \in \mathcal{R}^{(s+1)m \times n},$$

$$H_{u,s} = \begin{bmatrix} D & 0 & & \\ CB & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ CA^{s-1}B & \cdots & CB & D \end{bmatrix}, H_{w,s} = \begin{bmatrix} 0 & 0 & & \\ C & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ CA^{s-1} & \cdots & C & 0 \end{bmatrix},$$

where $H_{u,s} \in \mathcal{R}^{(s+1)m\times(s+1)p}$, $H_{w,s}W_{k,s} + V_{k,s}$ represents the influence of the noise vectors on the process output, and $W_{k,s}$, $V_{k,s}$ are as defined in (4.63)–(4.64).

Recall that a (steady) Kalman filter,

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K\left(y(k) - \hat{y}(k)\right), \hat{y}(k) = C\hat{x}(k) + Du(k),$$

can be re-written as

$$\begin{gathered}\hat{x}(k+1) = A_K\hat{x}(k) + B_K u(k) + Ky(k), \\ A_K = A - KC, B_K = B - KD\end{gathered} \tag{4.66}$$

with K as the Kalman filter gain matrix. It yields, for some integers i, s_p,

$$\begin{gathered}\hat{x}(k-i) = A_K^{s_p}\hat{x}(k-i-s_p) + \left[\, A_K^{s_p-1}B_K \cdots B_K \,\right] u_{s_p-1}(k-i-1) \\ + \left[\, A_K^{s_p-1}K \cdots K \,\right] y_{s_{p-1}}(k-i-1) \Longrightarrow \\ \hat{X}_{k-s} = \left[\, \hat{x}(k-s) \cdots \hat{x}(k-s+N-1) \,\right] \\ = A_K^{s_p}\left[\, \hat{x}(k-s-s_p) \cdots \hat{x}(k-s+N-1-s_p) \,\right] + L_p Z_p, \\ L_p = \left[\, A_K^{s_p-1}B_K \cdots B_K \; A_K^{s_p-1}K \cdots K \,\right], Z_p = \begin{bmatrix} U_{k-s-1,s_p-1} \\ Y_{k-s-1,s_p-1} \end{bmatrix}, \\ U_{k-s-1,s_p-1} = \left[\, u_{s_p-1}(k-s-1) \cdots u_{s_p-1}(k-s-2+N) \,\right], \\ Y_{k-s-1,s_p-1} = \left[\, y_{s_{p-1}}(k-s-1) \cdots y_{s_{p-1}}(k-s-2+N) \,\right].\end{gathered}$$

For a sufficiently large s_p, it is reasonable to assume

$$A_K^{s_p} \approx 0.$$

Substituting X_{k-s} by its estimation $\hat{X}_{k-s}$ leads to

$$Y_{k,s} = \Gamma_s L_p Z_p + H_{u,s}U_{k,s} + H_{w,s}W_{k,s} + V_{k,s}. \tag{4.67}$$

Note that only process input and output data as well as noises are included in (4.67). Hence, it is called I/O data model.

4.4.2 Identification of an SKR-based Residual Generator

Corresponding to Z_p, let

$$Z_f = \begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix}.$$

Here, the sub-indices of Z, f and p, stand for "future and past", respectively. It holds

$$Z_f = \begin{bmatrix} 0 & I \\ \Gamma_s L_p & H_{u,s} \end{bmatrix} \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix} + \begin{bmatrix} 0 \\ H_{w,s} W_{k,s} + V_{k,s} \end{bmatrix}. \tag{4.68}$$

The residual generation problem can then be formulated as finding null matrix $\mathcal{K}$ so that

$$\mathcal{K} \begin{bmatrix} 0 & I \\ \Gamma_s L_p & H_{u,s} \end{bmatrix} = 0. \tag{4.69}$$

In other words, in the noise-free case,

$$\mathcal{K} Z_f = \mathcal{K} \begin{bmatrix} 0 & I \\ \Gamma_s L_p & H_{u,s} \end{bmatrix} \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix} = 0.$$

$\mathcal{K}$ solving (4.69) is also called data-driven realisation of the kernel representation of system (4.61)–(4.62). Below, we briefly present an approach to finding $\mathcal{K}$.

By an LQ decomposition of the data sets,

$$\begin{bmatrix} Z_p \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix},$$

where

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \begin{bmatrix} Q_1^T & Q_2^T & Q_3^T \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix},$$

it turns out

$$Z_f = \begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} + \begin{bmatrix} 0 \\ L_{33} \end{bmatrix} Q_3.$$

Note that due to the whiteness of the noises, for a (very) large N

$$\frac{1}{N} \left[H_{w,s} W_{k,s} + V_{k,s} \right] \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix}^T \approx 0,$$

and moreover,

$$\begin{aligned} Z_f \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix}^T &= \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & I \\ \Gamma_s L_p & H_{u,s} \end{bmatrix} \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix} \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix}^T. \end{aligned}$$

Hence,

$$L_{33}Q_3 = H_{w,s}W_{k,s} + V_{k,s}, \tag{4.70}$$

which means, in turn,

$$\mathcal{K}\begin{bmatrix} 0 & I \\ \Gamma_s L_p & H_{u,s} \end{bmatrix} = 0 \Longleftrightarrow \mathcal{K}\begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} = 0. \tag{4.71}$$

Let

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 \end{bmatrix}$$

solve (4.71). It is evident that

$$\mathcal{K}_2\Gamma_s L_p = 0, \mathcal{K}_2 H_{u,s} = -\mathcal{K}_1. \tag{4.72}$$

Note that it is reasonable to assume that, for a large $s_p(>> n)$,

$$rank\left(L_p\right) = n.$$

As a result, $\mathcal{K}_2$ is in fact a parity matrix, as defined in (4.25), and the residual generator,

$$r(k) = \mathcal{K}_2 y_s(k) + \mathcal{K}_1 u_s(k) = \mathcal{K}_2\left(y_s(k) - H_{u,s}u_s(k)\right) \tag{4.73}$$

is equivalent to the parity space residual generator given in (4.24). It is worth noticing that

- on the assumption

  $$rank\left(\Gamma_s\right) = n$$

  the maximum rank of $\mathcal{K}_2$ is $(s+1)m - n$, which means in turn

  $$r(k) \in \mathcal{R}^{(s+1)m-n};$$

- the residual generator (4.72) can be parameterised by

  $$r(k) = P\mathcal{K}_2 y_s(k) + P\mathcal{K}_1 u_s(k), P \in \mathcal{R}^{((s+1)m-n)\times(s+1)m}, \tag{4.74}$$

  since $\forall P \in \mathcal{R}^{((s+1)m-n)\times(s+1)m}$,

  $$\mathcal{K}\begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} = 0 \Longrightarrow P\mathcal{K}\begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} = 0.$$

Suppose that an m-dimensional residual vector is generated by means of

$$r(k) = P_m\mathcal{K}_2 y_s(k) + P_m\mathcal{K}_1 u_s(k) \in \mathcal{R}^m, P_m \in \mathcal{R}^{m\times(s+1)m}.$$

Let

$$P_m \mathcal{K}_2 = \left[\bar{V}_0 \cdots \bar{V}_s \right], \bar{V}_i \in \mathcal{R}^{m \times m}, i = 0, 1, \cdots, s,$$

and assume

$$rank \left(\bar{V}_s \right) = m.$$

Then, we can write $r(k)$ as, similar to the parameterisation form given in Remark 4.2,

$$\begin{aligned} r(k) &= V \left(y(k) - \hat{y}(k) \right), V = \bar{V}_s, \\ \hat{y}(k) &= -\bar{V}_s^{-1} \left(\bar{V} y_{s-1}(k-1) + P_m \mathcal{K}_1 u_s(k) \right), \\ \bar{V} &= \left[\bar{V}_0 \cdots \bar{V}_{s-1} \right]. \end{aligned}$$

4.4.3 An Alternative Algorithm of SKR Identification

Recall that

$$Y_{k,s} = \left[L_{31} \; L_{32} \right] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} + L_{33} Q_3$$

and $L_{33} Q_3$ represents the process and measurement noises. Hence,

$$\hat{Y}_{k,s} = \left[L_{31} \; L_{32} \right] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \left[L_{31} \; L_{32} \right] \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^+ \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix}$$

delivers an estimate for $Y_{k,s}$, which can then be used for generating the residual (matrix)

$$Y_{k,s} - \hat{Y}_{k,s} = Y_{k,s} - \left[L_{31} \; L_{32} \right] \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^+ \begin{bmatrix} Z_p \\ U_{k,s} \end{bmatrix}. \tag{4.75}$$

Here, $\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^+$ is pseudo-inverse of $\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}$. Let

$$\left[K_p \; K_{f,u} \right] = \left[L_{31} \; L_{32} \right] \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^+.$$

The residual form (4.75) can be re-written into

$$Y_{k,s} - \hat{Y}_{k,s} = Y_{k,s} - K_p Z_p - K_{f,u} U_{k,s}. \tag{4.76}$$

Correspondingly, the residual generator for the online residual generation is constructed by

$$r(k) = y_s(k) - K_p \begin{bmatrix} u_{s-1}(k-s-1) \\ y_{s-1}(k-s-1) \end{bmatrix} - K_{f,u} u_s(k). \tag{4.77}$$

Here, s_p is set to be s. In other words,

$$\hat{y}_s(k) := K_p \begin{bmatrix} u_{s-1}(k-s-1) \\ y_{s-1}(k-s-1) \end{bmatrix} + K_{f,u} u_s(k)$$

can be viewed as an estimate of $y_s(k)$.

4.4.4 Fault Detection

Once the residual generator (4.72) is identified, the remaining tasks for a successful fault detection are to define a test statistic and to determine the corresponding threshold. For our purpose, we assume that the process and measurement noises are normally distributed with

$$\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{matrix} \Sigma_w & S_{wv} \\ S_{wv}^T & \Sigma_v \end{matrix}\right).$$

Recall that in the fault-free case

$$\begin{aligned} r(k) &= \mathcal{K}_2 \left(y_s(k) - H_{u,s} u_s(k)\right) = \mathcal{K}_2 \theta_s(k), \\ \theta_s(k) &= H_{w,s} w_s(k) + v_s(k) \sim \mathcal{N}\left(0, \Sigma_\theta\right). \end{aligned}$$

Moreover, on the assumption that the number of the training data N is sufficiently large, it holds

$$\frac{1}{N} \left(H_{w,s} W_{k,s} + V_{k,s}\right) \left(H_{w,s} W_{k,s} + V_{k,s}\right)^T \approx \Sigma_\theta,$$

which, considering (4.70), can be re-written as

$$\frac{1}{N} L_{33} Q_3 Q_3^T L_{33}^T = \frac{1}{N} L_{33} L_{33}^T \approx \Sigma_\theta.$$

To detect a change in the mean of $r(k)$, it is optimal, as we have discussed in Sect. 3.1,

- to define the test statistic

$$J = r^T(k) N \left(\mathcal{K}_2 L_{33} L_{33}^T \mathcal{K}_2^T\right)^{-1} r(k), \tag{4.78}$$

which can be, for a sufficiently large N, assumed to be

$$J \sim \chi^2\left((s+1)m-n\right),$$

- and to set the threshold
$$J_{th} = \chi^2_\alpha((s+1)m-n). \tag{4.79}$$

It is worth noting that a large s results in high threshold setting and thus may significantly reduce the fault detectability. Furthermore, the residual vector $r(k)$ contains redundant information. These suggest to select a number of rows from $\mathcal{K}_2$ to build the test statistic. To this end, the following modifications can be adopted:

- Do an SVD on $\frac{1}{N}\mathcal{K}_2 L_{33} L_{33}^T \mathcal{K}_2^T$,
$$\frac{1}{N}\mathcal{K}_2 L_{33} L_{33}^T \mathcal{K}_2^T = U\Sigma U^T,\ \Sigma = diag\left(\sigma_1^2, \cdots, \sigma_{(s+1)m-n}^2\right);$$
- Form
$$\Sigma_2 = diag\left(\sigma_{l+1}^2, \cdots, \sigma_{(s+1)m-n}^2\right) \in \mathcal{R}^{((s+1)m-n-l)\times((s+1)m-n-l)}, l >> 1$$
and correspondingly U_2
$$U\Sigma U^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} diag\left(\sigma_1^2, \cdots, \sigma_l^2\right) U_1^T \\ diag\left(\sigma_{l+1}^2, \cdots, \sigma_{(s+1)m-n}^2\right) U_2^T \end{bmatrix};$$
- Set
$$r(k) = U_2^T \mathcal{K}_2\left(y_s(k) - H_{u,s} u_s(k)\right) \in \mathcal{R}^{(s+1)m-n-l}, \tag{4.80}$$
$$J = r^T(k) diag\left(\sigma_{l+1}^{-2}, \cdots, \sigma_{(s+1)m-n}^{-2}\right) r(k), \tag{4.81}$$
$$J_{th} = \chi^2_\alpha((s+1)m-n-l). \tag{4.82}$$

The idea behind this alternative solution is evident, as given below,

$$\begin{gathered} r(k) = U_2^T \mathcal{K}_2\left(y_s(k) - H_{u,s} u_s(k)\right) = U_2^T \mathcal{K}_2 \theta_s(k) \sim \mathcal{N}\left(0, U_2^T \mathcal{K}_2 \Sigma_\theta \mathcal{K}_2^T U_2\right), \\ U_2^T \mathcal{K}_2 \Sigma_\theta \mathcal{K}_2^T U_2 = U_2^T U \Sigma U^T U_2 = diag\left(\sigma_{l+1}^2, \cdots, \sigma_{(s+1)m-n}^2\right) \Longrightarrow \\ r(k) \sim \mathcal{N}\left(0, diag\left(\sigma_{l+1}^2, \cdots, \sigma_{(s+1)m-n}^2\right)\right), J \sim \chi^2\left((s+1)m-n-l\right). \end{gathered}$$

We would like to emphasise that the selected $((s+1)m-n-l)$ -dimensional residual vector corresponds the residual subspace with the $(s+1)m-n-l$ smallest singular values $\sigma_{l+1}^2, \cdots, \sigma_{(s+1)m-n}^2$, which, as discussed in Sect. 3.1, delivers the optimal fault detectability.

4.4.5 Observer-Based Implementation

In its original form (4.24), a parity space based residual generator is in fact a finite impulse response filter (FIRF). In control engineering, it is state of the art that control and detection systems are implemented in a recursive form, that is as an infinite impulse response filter (IIRF). For instance, an FDF is an IIRF. In this sub-section, we briefly introduce an approach, which allows us to realise a parity space based residual generator in form of a diagnostic observer.

Consider the process model (4.2)–(4.3) and a parity vector

$$v_s = \begin{bmatrix} v_{s,0} & v_{s,1} & \cdots & v_{s,s} \end{bmatrix} \in \mathcal{R}^{(s+1)m}, \; v_s \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix} = 0,$$

$v_{s,i} \in \mathcal{R}^m, i = 0, 1, \cdots, s$. It has been demonstrated that matrices

$$A_z = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathcal{R}^{s\times s}, \; L_z = -\begin{bmatrix} v_{s,0} \\ v_{s,1} \\ \vdots \\ v_{s,s-1} \end{bmatrix}, \tag{4.83}$$

$$T = \begin{bmatrix} v_{s,1} & v_{s,2} & \cdots & v_{s,s-1} & v_{s,s} \\ v_{s,2} & \cdots & \cdots & v_{s,s} & 0 \\ \vdots & \cdots & \cdots & \vdots & \vdots \\ v_{s,s} & 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} \tag{4.84}$$

solve the Luenberger equations

$$TA - A_z T = L_z C, c_z T = gC, c_z = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}, g = v_{s,s}. \tag{4.85}$$

A direct application of this result is the construction of an observer-based residual generator for the given parity vector v_s as follows

$$z(k+1) = A_z z(k) + B_z u(k) + L_z y(k) \in \mathcal{R}^s, B_z = TB - L_z D, \tag{4.86}$$

$$r(k) = gy(k) - c_z z(k) - d_z u(k) \in \mathcal{R}, d_z = gD. \tag{4.87}$$

Note that

$$\begin{bmatrix} B_z \\ d_z \end{bmatrix} = \begin{bmatrix} v_{s,0} & v_{s,1} & \cdots & v_{s,s-1} & v_{s,s} \\ v_{s,1} & \cdots & \cdots & v_{s,s} & 0 \\ \vdots & \cdots & \cdots & \vdots & \vdots \\ v_{s,s} & 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} D \\ CB \\ CAB \\ \vdots \\ CA^{s-1}B \end{bmatrix} \tag{4.88}$$

$$= \begin{bmatrix} v_s H_{u,s}(:, 1:p) \\ v_s H_{u,s}(:, p+1:2p) \\ \vdots \\ v_s H_{u,s}(:, sp+1:(s+1)p) \end{bmatrix}.$$

Equations (4.83), (4.85) and (4.88) allow a direct construction of a residual generator of form (4.86)–(4.87) using a row of the identified kernel matrix $\mathcal{K}$ and without any additional design effort. Concretely, let ψ_K be a row of $\mathcal{K}$ and of the form

$$\psi_K = \begin{bmatrix} \psi_{K_1} & \psi_{K_2} \end{bmatrix}.$$

Since ψ_{K_2} is a parity vector and

$$\psi_{K_1} = -\psi_{K_2} H_{u,s},$$

we have, besides of A_z, c_z given in (4.83) and (4.85), B_z, d_z, g, L_z formed in terms of ψ_{K_1}, ψ_{K_2}:

$$L_z = -\begin{bmatrix} \psi_{K_2}(1:m) \\ \vdots \\ \psi_{K_2}((s-1)m+1:sm) \end{bmatrix}, g = \psi_{K_2}((sm+1:(s+1)m), \tag{4.89}$$

$$B_z = -\begin{bmatrix} \psi_{K_1}(1:p) \\ \vdots \\ \psi_{K_1}((s-1)p+1:sp) \end{bmatrix}, d_z = -\psi_{K_1}(sp+1:(s+1)p). \tag{4.90}$$

It is well-known that an m-dimensional residual vector is necessary for a reliable fault detection and isolation in the framework of observer-based fault diagnosis systems. There are various schemes to extend the above result to the multiple case.

At the end of this section, we would like to summarise the major results:

- the data-driven approach introduced in this section leads to a direct identification of parity space generators as well as the statistic features of the residual vector,
- based on these results, a fault detection can be realised.
- Moreover, the one-to-one mapping between the parity space approach and the observer-based approach allows us to construct an observer-based residual generator using an identified parity vector.

4.5 Notes and References

In this chapter, we have reviewed the basics of the model-based fault detection methods for LTI systems. Most of the results can be found in the monographs [1–7] and in the early survey papers [8–11]. The first work on FDF and DO has been reported by Beard and Jones [12, 13], and Chow and Willsky have proposed the first optimal solution using the parity space scheme [14]. The unified solution for an integrated optimal design of FDF and threshold has been derived by Ding et al. [15]. The achieved optimal FDF is in fact an $\mathcal{H}_2$ observer [7]. It is remarkable that Theorem 4.1 reveals that the unified solution solves the optimal fault detection problem formulated in Definition 2.7.

Although it has not been addressed in our review, the topic of (optimal) indices based observer-based residual generator design has been widely investigated in the past decades. The $\mathcal{H}_\infty/\mathcal{H}_\infty$ design problem was first proposed and solved in [16], lately in [17–19]. In 1993, $\mathcal{H}_-/\mathcal{H}_\infty$ design problem was proposed and handled [20]. It has been first extensively studied after the publication of the LMI (linear matrix inequality) solution to this problem [21]. The most significant contributions to this issue are [21–27]. It is worth mentioning the work by Zhong et al. [28], Wang and Yang [29] and Chadli et al. [30], in which the $\mathcal{H}_-/\mathcal{H}_\infty$ design scheme has been applied to (i) uncertain LTI systems, (ii) the solution in a finite frequency range, and (iii) nonlinear systems modelled by means of T-S fuzzy technique, respectively. In [15, 31], it has been proved that the unified solution offers a simultaneous solution to the multi-objective $\mathcal{H}_i/\mathcal{H}_\infty$ optimisation problem, and the $\mathcal{H}_-/\mathcal{H}_\infty$, $\mathcal{H}_\infty/\mathcal{H}_\infty$ as well as $\mathcal{H}_-/\mathcal{H}_\infty$ in a finite frequency range are only special cases of the $\mathcal{H}_i/\mathcal{H}_\infty$ optimisation. The unified solution was derived using the co-inner-outer factorisation technique, which is, in comparison with the LMI solutions, computational less involved.

We have briefly introduced a mathematical and control theoretical tool, the factorisation technique, for the modelling and presentation of dynamic systems. The associated model forms like LCF and RCF will play an important role in our subsequent studies. An immediate application of the factorisation technique to residual generation is the parameterisation of all LTI residual generators that is expressed by (4.28)–(4.29), first proposed by Ding and Frank in 1990 [32]. The reader is referred to [7, 33] for more details. For instance, Lemma 4.1 can be found in [7].

SKR is an alternative model and residual generator form, which is widely adopted in our subsequent work, in dealing with not only LTI systems, but also time-varying and nonlinear systems. In fact, the SKR modelling is widely applied in research on nonlinear control systems [34].

In the second part of this chapter, we have summarised the data-driven fault detection methods for dynamic systems. It is the application of subspace identification technique (SIT) to the fault diagnosis study, first reported in [35–38]. SIT is well-established and widely applied in process identification [39–42]. A remarkable result is the data-driven forms of SKR and their identification, which have been introduced in [43–45] and recently extended by Li et al. [46] to a class of nonlinear systems. In our subsequent investigations, SKR will be applied to bridging the model-based and data-driven fault detection techniques.

The final result presented in this chapter on the observer-based implementation of parity vector based residual generators has been reported in [47] and applied for the purpose of *parity space design, observer-based implementation* [7] or for the data-driven design of observer-based residual generators [36, 43, 45].

References

1. J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York Basel Hong Kong: Marcel Dekker, 1998.
2. J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Boston: Kluwer Academic Publishers, 1999.
3. R. J. Patton, P. M. Frank, and R. N. C. (Eds.), *Issues of Fault Diagnosis for Dynamic Systems*. London: Springer, 2000.
4. F. Gustafsson, *Adaptive Filtering and Change Detection*. Chichester: John Wiley and Sons, LTD, 2000.
5. M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control, 2nd Edition*. Berlin Heidelberg: Springer, 2006.
6. S. Simani, S. Fantuzzi, and R. J. Patton, *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. London: Springer-Verlag, 2003.
7. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.
8. P. M. Frank and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of Process Control*, vol. 7(6), pp. 403–424, 1997.
9. P. Zhang and S. X. Ding, "On fault detection in linear discrete-time, periodic, and sampled-data systems (survey)," *Journal of Control Science and Engineering*, pp. 1–18, 2008.
10. R. Mangoubi, M. Desai, A. Edelmayer, and P. Sammak, "Robust detection and estimation in dynamic systems and statistical signal processing: Intersection, parallel paths and applications," *European Journal of Control*, vol. 15, pp. 348–369, 2009.
11. I. Hwang, S. Kim, Y. Kim, and C. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Trans. Contr. Syst. Tech.*, vol. 18, pp. 636–653, 2010.
12. R. Beard, *Failure Accomondation in Linear Systems Through Self-Reorganization*. PhD Dissertation, MIT, 1971.
13. H. Jones, *Failure Detection in Linear Systems*. PhD dissertation, MIT, 1973.
14. E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. on Automatic Control*, vol. 29, pp. 603–614, 1984.
15. S. X. Ding, T. Jeinsch, P. M. Frank, and E. L. Ding, "A unified approach to the optimization of fault detection systems," *International Journal of Adaptive Control and Signal Processing*, vol. 14, pp. 725–745, 2000.
16. X. Ding and P. M. Frank, "Frequency domain approach and threshold selector for robust model-based fault detection and isolation," *Proc. of the 1st IFAC Symp. SAFEPROCESS*, 1991.
17. Z. Qiu and J. J. Gertler, "Robust FDI systems and h-infinity optimization," *Proc. of ACC 93*, pp. 1710–1715, 1993.
18. P. M. Frank and X. Ding, "Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis," *Automatica*, vol. 30, pp. 789–904, 1994.
19. D. Sauter and F. Hamelin, "Frequency-domain optimization for robust fault detection and isolation in dynamic systems," *IEEE Trans. on Autom. Control*, vol. 44, pp. 878–882, 1999.
20. X. Ding, L. Guo, and P. M. Frank, "A frequency domain approach to fault detection of uncertain dynamic systems," in *Proc. of the 32nd Conference on Decision and Control*, Texas, USA, 1993, pp. 1722–1727.
21. M. Hou and R. J. Patton, "An LMI approach to infinity fault detection observers," in *Proceedings of the UKACC International Conference on Control*, 1996, pp. 305–310.
22. F. Rambeaux, F. Hamelin, and D. Sauter, "Robust residual generation via LMI," in *Proceedings of the 14th IFAC World Congress*, Beijing, China, 1999, pp. 240–246.
23. J. Liu, J. L. Wang, and G. H. Yang, "An LMI approach to minimum sensitivity analysis with application to fault detection," *Automatica*, vol. 41, pp. 1995–2004, 2005.
24. D. Henry and A. Zolghadri, "Design and analysis of robust residual generators for systems under feedback control," *Automatica*, vol. 41, pp. 251–264, 2005.

25. M. L. Rank and H. Niemann, "Norm based design of fault detectors," *International Journal of Control*, vol. 72(9), pp. 773–783, 1999.
26. J. L. Wang, G.-H. Yang, and J. Liu, "An LMI approach to H_- index and mixed H_-/H_{inf} fault detection observer design," *Automatica*, vol. 43, pp. 1656–1665, 2007.
27. Z. Li, E. Mazars, Z. Zhang, and I. M. Jaimoukha, "State-space solution to the h–/hinf fault detection problem," *Int. J. of Robst. Nonlinear Control*, vol. 22, pp. 282–299, 2012.
28. M. Zhong, S. X. Ding, J. Lam, and H. Wang, "An LMI approach to design robust fault detection filter for uncertain LTI systems," *Automatica*, vol. 39, pp. 543–550, 2003.
29. H. Wang and G.-H. Yang, "A finite frequency domain approach to fault detection observer design for linear continuous-time systems," *Asian Journal of Control*, vol. 10, pp. 1–10, 2008.
30. M. Chadli, A. Abdo, and S. X. Ding, "H_-/H_{inf} fault detection filter design for discrete-time takagi-sugeno fuzzy system," *Automatica*, vol. 49, pp. 1996–2005, 2013.
31. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms, and Tools*. Springer-Verlag, 2008.
32. X. Ding and P. M. Frank, "Fault detection via factorization approach," *Syst. and Contr. Letters*, vol. 14, pp. 431–436, 1990.
33. K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, New Jersey: Prentice-Hall, 1996.
34. A. Van der Schaft, *L2 - Gain and Passivity Techniques in Nonlinear Control*. London: Springer, 2000.
35. S. J. Qin and W. Li, "Detection and identification of faulty sensors in dynamic processes," *AIChE Journal*, vol. 47, pp. 1581–1593, 2001.
36. S. X. Ding, P. Zhang, A. Naik, E. Ding, and B. Huang, "Subspace method aided data-driven design of fault detection and isolation systems," *Journal of Process Control*, vol. 19, pp. 1496–1510, 2009.
37. J. Dong and M. Verhaegen, "Subspace based fault detection and identification for LTI systems," *Proc. of the 7th IFAC Symp. SAFEPROCESS*, pp. 330–335, 2009.
38. J. Dong, "Data driven fault tolerant control: A subspace approach," Ph.D. dissertation, Technische Universiteit Delft, 2009.
39. W. Favoreel, B. D. Moor, and P. V. Overschee, "Subspace state space system identification for industrial processes," *Journal of Process Control*, vol. 10, pp. 149–155, 2000.
40. P. V. Overschee and B. D. Moor, *Subspace Identification for Linear Systems*. USA: Kluwer Academic Publishers, 1996.
41. S. J. Qin, "An overview of subspace identification," *Computers and Chemical Engineering*, vol. 30, pp. 1502–1513, 2006.
42. B. Huang and R. Kadali, *Dynamic Modelling, Predictive Control and Performance Monitoring, a Data-Driven Subspace Approach*. London: Springer-Verlag, 2008.
43. S. X. Ding, "Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results," *Journal of Process Control*, vol. Vol. 24, pp. 431–449, 2014.
44. S. X. Ding, Y. Yang, Y. Zhang, and L. Li, "Data-driven realization of kernel and image representations and their application to fault detection and control system design," *Automatica*, vol. 50, pp. 2615–2623, 2014.
45. S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems*. London: Springer-Verlag, 2014.
46. L. Li, S. X. Ding, Y. Yang, K. Peng, and J. Qiu, "A fault detection approach for nonlinear systems based on data-driven realizations of fuzzy kernel representations," *IEEE Trans. on Fuzzy systems*, vol. 26, pp. 1800–1812, 2018.
47. X. Ding, L. Guo, and T. Jeinsch, "A characterization of parity space and its application to robust fault detection," *IEEE Trans. on Automatic Control*, vol. 44(2), pp. 337–343, 1999.

Chapter 5
Feedback Control, Observer and Residual Generation

Fault-tolerant control is one of the major topics of this book. It is state of the art that fault-tolerant control is generally dealt with in the context of accommodating or/and re-configuring an operating controller to maintain reliable and fail-safe system operations, when faults are detected and identified in the system. Roughly speaking, a fault-tolerant control scheme is implemented in two steps:

- a fault diagnosis system is running real-time, synchronised with the process operation, and activates the fault-tolerant action, when a fault is detected and identified,
- the controller is then accommodated or re-configured based on information about the fault received from the diagnosis system.

Consequently, a fault-tolerant control system is often designed in two separate units: a fault diagnosis system and an accommodatable and re-configurable control system.

On the other hand, recent investigations reveal that feedback control and fault detection share the process information presented in form of residual signals. This fact allows us to design fault-tolerant control systems in an integrated manner, that is, integrated design of the fault diagnosis and control systems. The expected benefit of such an integrated design is improvement in the system efficiency and performance. Most of the fault-tolerant control schemes addressed and developed in this book are based on the principle of integrated design. This also motivates us to review and introduce needed preliminaries and results in this chapter. For our purpose, we will attempt to study and provide "residual relevant" insights and interpretations of the well-established feedback control theoretical framework.

5.1 Preliminaries

We first introduce preliminary knowledge needed for the feedback controller configuration and design, which builds the basis for our fault-tolerant control architecture and the relevant study. We consider the nominal model

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_5

$$x(k+1) = Ax(k) + Bu(k), x(0) = x_0, \tag{5.1}$$
$$y(k) = Cx(k) + Du(k). \tag{5.2}$$

5.1.1 State Feedback Control, RCF and Image Representation

Recall that in Chap. 4 we have introduced LCF, RCF as well as SKR, where RCF is defined as follows. Let

$$G_{yu}(z) = C(zI - A)^{-1}B + D.$$

The pair $(M(z), N(z))$,

$$\begin{aligned} M(z) &= I + F\left(zI - A_F\right)^{-1}B, N(z) = D + C_F\left(zI - A_F\right)^{-1}B, \\ A_F &= A + BF, C_F = C + DF, \end{aligned}$$

builds the RCF of $G_{yu}(z)$ with

$$G_{yu}(z) = N(z)M^{-1}(z),$$

where F is a matrix of appropriate dimension and A_F is a Schur matrix. It is well-known that the interpretation of RCF is state feedback control with

$$\begin{aligned} x(k+1) &= (A+BF)\,x(k) + B\upsilon(k), y(k) = (C+DF)\,x(k) + D\upsilon(k), \\ u(k) &= Fx(k) + \upsilon(k) \Longrightarrow u(z) = M(z)\upsilon(z), y(z) = N(z)\upsilon(z), \end{aligned} \tag{5.3}$$

and $\upsilon(z)$ being the reference vector. Alternatively, the nominal model (5.1)-(5.2) can be represented by

$$\text{for some } \upsilon \in \mathcal{H}_2, \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \upsilon(z). \tag{5.4}$$

As a dual form of the SKR introduced in Definition 4.3, (5.4) is called (stable) image representation of system (5.1)–(5.2).

Definition 5.1 *Given system (5.1)–(5.2), then a stable linear system $\mathcal{I}$ is called stable image representation (SIR) of (5.1)–(5.2), when for any $u(z)$ and its response $y(z)$ a (reference) input $\upsilon(z)$ can be found such that*

$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \mathcal{I}\upsilon(z). \tag{5.5}$$

A direct application of SIR is to design feed-forward controllers. Suppose that $\upsilon(z)$ is driven by a reference signal $y_{ref}(z)$ to be followed by the plant output $y(z)$. Let

$$\upsilon(z) = T(z)y_{ref}(z) \Longrightarrow N(z)\,\upsilon(z) = N(z)\,T(z)y_{ref}(z). \tag{5.6}$$

$T(z)$ is a feed-forward controller, which can be applied to approaching the desired tracking behaviour. It yields

$$y(z) = N(z)T(z)y_{ref}(z).$$

Moreover, the dynamic relation between $\upsilon(z)$ and $u(z)$,

$$u(z) = M(z)\upsilon(z),$$

models the actuator dynamics, which can be used, for instance, for the purpose of actuator monitoring.

5.1.2 Parameterisation of all Stabilising Controllers

Consider the feedback control loop sketched in Fig. 5.1 with plant model $G_{yu}(z)$ and controller $K(z)$. Suppose that the model (5.1)–(5.2) is the minimal state space realisation of $G_{yu}(z)$ and, associated with it, there are eight transfer matrices, left and right coprime pairs of $G_{yu}(z)$, $\left(\hat{M}(z), \hat{N}(z)\right)$ and $(M(z), N(z))$, as well as the other two coprime pairs $\left(\hat{X}(z), \hat{Y}(z)\right)$ and $(X(z), Y(z))$, as given in (4.7)–(4.10). The so-called Youla parameterisation described by

$$K(z) = -\left(X(z) - Q_c(z)\hat{N}(z)\right)^{-1}\left(Y(z) + Q_c(z)\hat{M}(z)\right) \tag{5.7}$$

$$= -\left(\hat{Y}(z) + M(z)Q_c(z)\right)\left(\hat{X}(z) - N(z)Q_c(z)\right)^{-1} \tag{5.8}$$

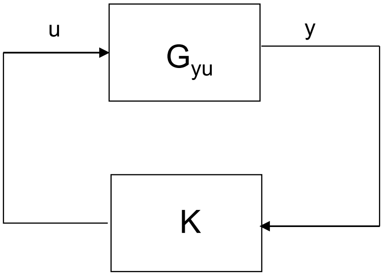


Fig. 5.1 Feedback control loop

parameterises all stabilising controllers by the stable parameter matrix $Q_c(z)$. In other words, the feedback control loop is stable if the controller $K(z)$ is expressed either in form of (5.7) or (5.8), and vice versa.

Note that for the controller

$$u(z) = K(z)y(z),$$

the pairs,

$$\left(-X(z) + Q_c(z)\hat{N}(z), Y(z) + Q_c(z)\hat{M}(z)\right),$$
$$\left(-\hat{Y}(z) - M(z)Q_c(z), \hat{X}(z) - N(z)Q_c(z)\right),$$

build the LCF and RCF of $K(z)$, which can also be written in form of SKR and SIR of the controller as follows:

$$\left[\, X(z) - Q_c(z)\hat{N}(z) \;\; Y(z) + Q_c(z)\hat{M}(z) \,\right] \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = 0,$$
$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} -\hat{Y}(z) - M(z)Q_c(z) \\ \hat{X}(z) - N(z)Q_c(z) \end{bmatrix} v(z).$$

5.2 On Bezout Identity and Parameterisation of Stabilising Controllers

5.2.1 Observer-Based Realisations of Feedback Controllers

The control theoretical interpretations of the RCF and LCF, namely, the state feedback controller and observer-based residual generator, have been briefly introduced in the previous sections. It is of considerable interest to reveal the control theoretical interpretations of the other four transfer matrices, $X(z), Y(z), \hat{X}(z), \hat{Y}(z)$, in the Bezout identity with the state space representations given in (4.9)–(4.10).

We first consider $\hat{X}(z), -\hat{Y}(z)$. It is evident that they are, due to Bezout identity (4.11), right coprime. Next, re-write observer-based residual generator (4.12)–(4.13) as

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr(k), r(k) = y(k) - C\hat{x}(k) - Du(k). \tag{5.9}$$

Let

$$u(k) = F\hat{x}(k)$$

be an observer-based state feedback controller. Substituting it into the residual generator yields

$$\hat{x}(k+1) = (A + BF)\,\hat{x}(k) + Lr(k),\, y(k) = r(k) + (C + DF)\,\hat{x}(k). \tag{5.10}$$

Comparing with (4.9), it becomes clear that

$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} -\hat{Y}(z) \\ \hat{X}(z) \end{bmatrix} r(z). \tag{5.11}$$

That is, $\hat{X}(z), -\hat{Y}(z)$ are an observer-based system with the residual vector as its input and $u(k), y(k)$ as its output. It can also be understood as an SIR of the dynamic output controller

$$u(z) = K(z)y(z) = -\hat{Y}(z)\hat{X}^{-1}(z)y(z). \tag{5.12}$$

We now consider $X(z), -Y(z)$. Re-writing the observer (5.9) as

$$\hat{x}(k+1) = (A - LC)\,\hat{x}(k) + (B - LD)\,u(k) + Ly(k),$$

it is straightforward that for

$$u(k) = F\hat{x}(k) + v(k) \Longleftrightarrow v(z) = u(z) - F\hat{x}(z),$$

it holds

$$X(z)u(z) + Y(z)y(z) = v(z). \tag{5.13}$$

Hence, $X(z), -Y(z)$ are in fact an observer-driven system. For $v(z) = 0$,

$$\begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = 0$$

builds an SKR of the controller

$$u(z) = K(z)y(z) = -X^{-1}(z)Y(z)y(z), \tag{5.14}$$

and thus it is also an LCF of $K(z)$.

5.2.2 Bezout Identity and Feedback Control Loops

It is interesting to notice that by means of (5.11) and (5.13) as well as (5.12) and (5.14), the proof of Bezout identity (4.11) becomes evident. Moreover, the Bezout identity can be interpreted as the following system realisation: Given the system

(5.1)–(5.2) with $x_0 \neq 0$, and re-write it as

$$x(k+1) = Ax(k) + Bu(k) + \bar{x}_0\delta(k), x(0) = 0, \bar{x}_0 = Ax_0, \tag{5.15}$$

$$y(k) = Cx(k) + Du(k), \delta(k) = \begin{cases} 1, k = 0, \\ 0, k \neq 0. \end{cases} \tag{5.16}$$

For our purpose, system (5.15)–(5.16) is decomposed into two sub-systems

$$x(k) = x_1(k) + x_2(k), y(k) = y_1(k) + y_2(k), u(k) = u_1(k) + u_2(k)$$

with

$$x_1(k+1) = Ax_1(k) + Bu_1(k), x_1(0) = 0, \tag{5.17}$$

$$y_1(k) = Cx_1(k) + Du_1(k), u_1(k) = Fx_1(k) + v(k), \tag{5.18}$$

$$x_2(k+1) = Ax_2(k) + Bu_2(k) + \bar{x}_0\delta(k), x_2(0) = 0, \tag{5.19}$$

$$y_2(k) = Cx_2(k) + Du_2(k), u_2(k) = F\hat{x}_2(k), \tag{5.20}$$

$$\hat{x}_2(k+1) = (A + BF)\,\hat{x}_2(k) + Lr(k), \tag{5.21}$$

$$r(k) = y_2(k) - C\hat{x}_2(k) - Du_2(k). \tag{5.22}$$

As a result, we have the following system dynamics,

$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} u_1(z) + u_2(z) \\ y_1(z) + y_2(z) \end{bmatrix} = \begin{bmatrix} M(z) & -\hat{Y}(z) \\ N(z) & \hat{X}(z) \end{bmatrix} \begin{bmatrix} v(z) \\ r(z) \end{bmatrix}. \tag{5.23}$$

On the other hand, note that

$$x_1(k+1) = Ax_1(k) + Bu_1(k) = (A - LC)\,x_1(k) + (B - LD)\,u_1(k) + Ly_1(k),$$
$$u_1(k) = Fx_1(k) + v(k) \Longleftrightarrow v(k) = u_1(k) - Fx_1(k)$$

is a state space realisation of

$$v(z) = \begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} u_1(z) \\ y_1(z) \end{bmatrix}.$$

Moreover,

$$\begin{bmatrix} u_2(z) \\ y_2(z) \end{bmatrix} = \begin{bmatrix} -\hat{Y}(z) \\ \hat{X}(z) \end{bmatrix} r(z), \begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} -\hat{Y}(z) \\ \hat{X}(z) \end{bmatrix} = 0.$$

It holds

$$v(z) = \begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} u_1(z) + u_2(z) \\ y_1(z) + y_2(z) \end{bmatrix} = \begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix}.$$

Similarly, we have

$$\begin{aligned}\begin{bmatrix}-\hat{N}(z) & \hat{M}(z)\end{bmatrix}\begin{bmatrix}u(z)\\ y(z)\end{bmatrix} &= \begin{bmatrix}-\hat{N}(z) & \hat{M}(z)\end{bmatrix}\begin{bmatrix}u_1(z)+u_2(z)\\ y_1(z)+y_2(z)\end{bmatrix}\\ &= \begin{bmatrix}-\hat{N}(z) & \hat{M}(z)\end{bmatrix}\begin{bmatrix}M(z)\\ N(z)\end{bmatrix}v(z)+r(z)=r(z).\end{aligned}$$

Thus, it finally results in

$$\begin{bmatrix}v(z)\\ r(z)\end{bmatrix}=\begin{bmatrix}X(z) & Y(z)\\ -\hat{N}(z) & \hat{M}(z)\end{bmatrix}\begin{bmatrix}u(z)\\ y(z)\end{bmatrix}. \tag{5.24}$$

This demonstrates that

$$\begin{bmatrix}M(z) & -\hat{Y}(z)\\ N(z) & \hat{X}(z)\end{bmatrix}^{-1}=\begin{bmatrix}X(z) & Y(z)\\ -\hat{N}(z) & \hat{M}(z)\end{bmatrix},$$

which gives a control-loop interpretation of Bezout identity (4.11). It is worth noticing that $\hat{x}_2(k)$ can be understood as an estimation of the change in the state vector, which is caused by the disturbance $\delta(k)$. Indeed, our interpretation can also be extended in a more general form:

$$x(k+1)=Ax(k)+Bu(k)+Ed(k), x(0)=0, \tag{5.25}$$

$$y(k)=Cx(k)+Du(k), \tag{5.26}$$

where $d(k)$ represents an unknown input vector, and $\bar{x}_0\delta(k)$ in (5.17)–(5.22) is substituted by $Ed(k)$. It should be emphasised that the residual vector $r(k)$ reflects the change in the system caused by the unknown input vector $d(k)$.

From the control system point of view, the system one with (u_1, y_1) as the input and output pair represents the response of a state feedback control loop to the reference signal, while the system two with (u_2, y_2) delivers the response to the uncertainty caused by the unknown initial condition under the use of an observer-based state feedback controller. The uncertainty is the driver of the residual vector. In other words, the residual signal is an indicator for the uncertainty in the feedback control loop. The above study and relation (5.23) reveal that the core of a feedback control is the feedback of the residual signal aiming at reducing the influence of the uncertainty on the feedback control loop.

5.2.3 Parameterisation of Bezout Identity

Using the identities included in the Bezout identity (4.11), the Bezout identity can be extended to

$$\begin{bmatrix} X(z) - Q(z)\hat{N}(z) & Y(z) + Q(z)\hat{M}(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix} \begin{bmatrix} M(z) & -\hat{Y}(z) - M(z)Q(z) \\ N(z) & \hat{X}(z) - N(z)Q(z) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

for any $Q(z)$ or as

$$\begin{bmatrix} X(z) & Y(z) \\ S(z)X(z) - \hat{N}(z) & \hat{M}(z) + S(z)Y(z) \end{bmatrix} \begin{bmatrix} M(z) + \hat{Y}(z)S(z) & -\hat{Y}(z) \\ N(z) - \hat{X}(z)S(z) & \hat{X}(z) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

for any $S(z)$. $Q(z), S(z)$ are called parameterisation matrices. In our subsequent study, they are assumed to belong to $\mathcal{RH}_\infty$.

Below, we study the Bezout identity parameterised by $Q(z)$. To this end, we consider again the system model (5.1)–(5.2) and the corresponding observer-based residual generator (5.9) with a controller

$$u(z) = F\hat{x}(z) - Q(z)r(z).$$

It turns out

$$\begin{aligned} \hat{x}(k+1) &= (A + BF)\,\hat{x}(k) + B\bar{r}(k) + Lr(k), \bar{r}(z) = -Q(z)r(z), \\ y(k) &= r(k) + (C + DF)\,\hat{x}(k) + D\bar{r}(k). \end{aligned}$$

Recalling (4.8)–(4.9), it leads to

$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} r(z). \tag{5.27}$$

Since $\left(-\hat{Y}(z) - M(z)Q(z), \hat{X}(z) - N(z)Q(z)\right)$ is a right coprime pair, we finally have

$$u(z) = -\left(\hat{Y}(z) + M(z)Q(z)\right)\left(\hat{X}(z) - N(z)Q(z)\right)^{-1} y(z).$$

In a similar way and analogue to our study in the last sub-section, it can also be proved that

$$\left(X(z) - Q(z)\hat{N}(z\right) u(z) + \left(Y(z) + Q(z)\hat{M}(z)\right) y(z) = v(z). \tag{5.28}$$

In this way, the Youla parameterisation forms (5.7) and (5.8) for all stabilising controllers are also demonstrated. They are observer-based systems.

5.3 An Observer-Based Fault-Tolerant Control Architecture

In this section, we focus on the observer-based realisation of the Youla parameterisation of all stabilising controllers, as we have demonstrated in the last section. This realisation form is the basis of the so-called observer-based fault-tolerant control architecture and will play a central role in our subsequent work on fault-tolerant control issues. For our purpose, we first summarise this result in form of a theorem.

Theorem 5.1 *Given the feedback control loop sketched in Fig. 5.1 with the plant model $G_{yu}(z)$ whose minimal state space realisation is given by (5.1)–(5.2), then all stabilising controllers can be realised by the following observer-based system:*

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr(k) \tag{5.29}$$

$$= (A - LC)\,\hat{x}(k) + (B - LD)\,u(k) + Ly(k), \tag{5.30}$$

$$u(z) = F\hat{x}(z) + Q(z)r(z),\; Q(z) \in \mathcal{RH}_\infty, \tag{5.31}$$

$$r(z) = y(k) - \hat{y}(k) = y(k) - C\hat{x}(k) - Du(k), \tag{5.32}$$

where $Q(z)$ is the parameterisation matrix.

5.3.1 An Output Feedback Controller is an Estimator

Given the system model (5.1)–(5.2), it has been demonstrated that system

$$\eta(z) = \left(F(z)X(z) - Q_o(z)\hat{N}(z)\right)u(z) + \left(F(z)Y(z) + Q_o(z)\hat{M}(z)\right)y(z), \tag{5.33}$$

$$F(z) = F\left(zI - A - BF\right)^{-1} B \in \mathcal{RH}_\infty,\; Q_o(z) \in \mathcal{RH}_\infty, \tag{5.34}$$

describes a parameterisation of all observers that deliver an estimation for $Fx(k)$ satisfying

$$\forall x(0), u(k),\;\; \lim_{k\to\infty} (\eta(k) - Fx(k)) = 0, \tag{5.35}$$

where $X(z)$, $Y(z)$, $\hat{N}(z)$, $\hat{M}(z)$ are the stable transfer function matrices given in (4.7) and (4.10), and $Q_o(z)$ is a parameterisation matrix. Moreover, (5.33) can be re-written into

$$\hat{x}(k+1) = (A - LC)\,\hat{x}(k) + (B - LD)\,u(k) + Ly(k), \tag{5.36}$$

$$\eta(z) = F\hat{x}(z) + R(z)\left(y(z) - \hat{y}(z)\right), \hat{y}(k) = C\hat{x}(k) + Du(k), \tag{5.37}$$

$$R(z) = Q_o(z) + \hat{Y}(z) \in \mathcal{RH}_\infty. \tag{5.38}$$

We now compare the above system with the output feedback controller (5.29)–(5.32) given in Theorem 5.1 and see clearly that this output feedback controller is indeed an observer-based estimator for $Fx(k)$. As a result, we claim, according to Theorem 5.1, that all stabilising output feedback controllers are an observer-based estimator.

This result reveals an important aspect of an output feedback controller: for a given state feedback gain matrix F, the performance of the controller depends on the estimation performance of the observer (5.36)–(5.37). In a certain sense, this aspect can be understood as an extension of the well-known separation principle.

5.3.2 A Fault-Tolerant Control Architecture

An immediate and obvious application of the observer-based realisation of all stabilising controllers described in Theorem 5.1 is the establishment of the fault-tolerant control system architecture sketc.hed in Fig. 5.2. It is composed of three functional modules:

- an observer and an observer-based residual generator,

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr(k),$$
$$r(k) = y(k) - \hat{y}(k), \hat{y}(k) = C\hat{x}(k) + Du(k),$$

  which serve as an information provider for the controller and diagnostic system and deliver a state estimation, $\hat{x}$, and the preliminary residual, $r = y - \hat{y}$,
- controllers
$$u(z) = F\hat{x}(z) + Q(z)r(z) + V(z)v(z),$$

  including

  - a feedback controller: $F\hat{x}(z) + Q(z)r(z)$ and
  - a feed-forward controller: $V(z)v(z)$,

- diagnostic residual generator $R(z)r(z)$, which is used for the fault diagnosis purpose.

We call the above three functional modules the low level components of a fault-tolerant control system, which run real-time during process operations. For a successful fault-tolerant control, further functional modules and algorithms like fault detection and identification algorithms, system re-configuration and adaptation will be integrated into the architecture. We call them high level functional modules, since

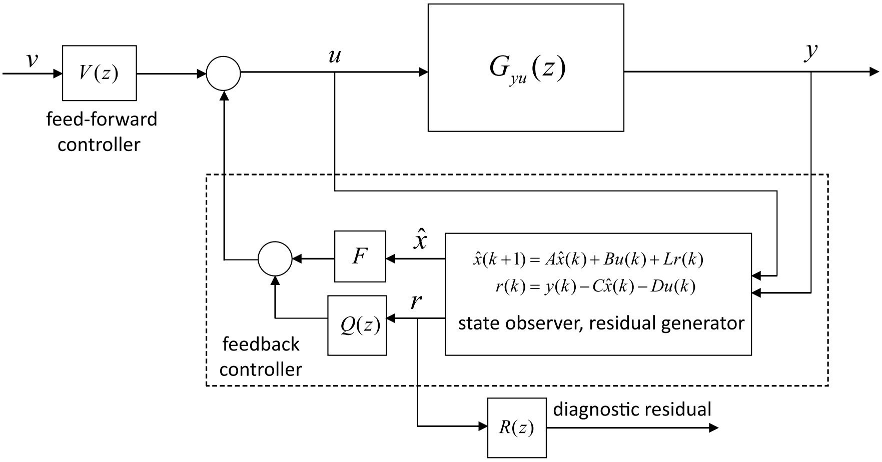


Fig. 5.2 A fault-tolerant control architecture

they are driven or activated by the diagnostic system at the low level and accommodate or re-configure the controllers, the observer as well as the diagnostic system.

The fault-tolerant control architecture is a platform, on which control and fault diagnosis are realised in an integrated manner with the observer as their core. From the control theoretical point of view, this fault-tolerant control architecture has the advantage that all dynamic systems integrated in the architecture are stable and the closed loop is well-posed. In particular, the modular structure provides us with

- clear parameterisations of the functional modules:
  - the state observer is parameterised by L,
  - the feedback controller by F, $Q(z)$,
  - the feed-forward controller by $V(z)$, and
  - the diagnostic residual generator by $R(z)$;
- functionalisation of the system parameters and
- prioritisation.

The last two properties are of special importance for a successful fault-tolerant control, as shortly described below.

Functionalisation Although all five parameters listed above are available in the fault-tolerant control architecture for the design and online optimisation objectives, they have evidently different functionalities, as summarised below:

- F, L determine the stability and eigen-dynamics of the closed-loop,
- $R(z)$, $V(z)$ have no influence on the system stability, and
- $Q(z)$ is used to enhance the system robustness and fault-tolerant performance. The design and modification of $Q(z)$ will affect the system stability, when parameter uncertainties or faults are present in the system,

- $R(z)$ serves for the optimisation of the fault detectability, and
- $V(z)$ for the tracking behaviour.

Prioritisation In the context of fault-tolerant control, the above five parameters have to be, due to their different functionalities, treated with different priorities. Recall that system stability is the minimum requirement on an automatic control system. This requires that a real-time adaptation of F, L to the possible changes in the system eigen-dynamics, possibly caused by faults, is to be performed, in order to guarantee the overall system stability. For this reason, adaptation of F, L should have the highest priority during the system operation. Differently, $Q(z)$, $R(z)$, $V(z)$ are used to optimise control or diagnosis performance. In case that a temporary system performance degradation is tolerable, the real-time demand and the priority for the optimisation of $Q(z)$, $R(z)$, $V(z)$ are relatively low.

5.4 Notes and References

This chapter is dedicated to the introduction of preliminary knowledge of feedback controller and observer design, which is needed for our study on fault-tolerant control and integrated design of control and diagnostic systems.

At first, we have reviewed the RCF in the control theoretic context and illustrated its interpretation as a state feedback controller. As a dual form to SKR, we have further introduced the SIR which is similar to the one given by [1] for nonlinear systems. SIR is a useful tool for the design of feedback and feed-forward controllers.

Youla parameterisation of all (LTI) stabilizing controllers is essential in robust control and for fault-tolerant controller design. Both the original form and its observer-based form can be found in [2, 3]. In this regard, we have revealed the following interesting aspects of Youla parameterisation of stabilizing controllers:

- any stabilising controller is an observer for the estimation of $Fx(k)$. This result has been demonstrated in [4, 5] based on the parameterisation of all LTI observers [6];
- the SIR of a stabilising controller is driven by the residual vector, and
- the core of a stabilising controller is the feedback of the residual signal aiming to reduce the influence of the uncertainty on the feedback control loop.

On the basis of the observer-based realisation of stabilising controllers, a fault-tolerant control architecture with an observer-based residual generator in its core is introduced and sketched in Fig. 5.2. This kind of fault-tolerant control architecture was initiated by [7] and extensively investigated in [4]. There are two aspects that may call the reader's attention. Firstly, this architecture can be used for the integrated design of control and diagnostic systems, which was first investigated by Nett et al. in 1988 [8] and intensively addressed later in [9–11]. The main idea of the integrated design scheme is to formulate the design of the controller and diagnostic system uniformly as a standard optimisation problem. Secondly, this fault-tolerant control architecture is helpful for us to gain a deeper insight into the design and (real-time) optimisation of feedback controllers. For instance, it enables a clear functionalisation of all controller parameters, and, consequently, helps us to realise a prioritisation of all the involved parameters. The latter is of significant importance for the online optimisation of the controller parameters by performing fault-tolerant control.

References

[1] A. Van der Schaft, *L2 - Gain and Passivity Techniques in Nonlinear Control*. London: Springer, 2000.

[2] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, New Jersey: Prentice-Hall, 1996.

[3] B. D. O. Anderson, "From youla-kucera to identification, adaptive and nonlinear control," *Automatica*, vol. 34, pp. 1485–1506, 1998.

[4] S. X. Ding, G. Yang, P. Zhang, E. Ding, T. Jeinsch, N. Weinhold, and M. Schulalbers, "Feedback control structures, embedded residual signals and feedcak control schemes with an integrated residual access," *IEEE Trans. on Contr. Syst. Tech.*, vol. 18, pp. 352–367, 2010.

[5] S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems*. London: Springer-Verlag, 2014.

[6] X. Ding, L. Guo, and P. M. Frank, "Parametrization of linear observers and its application to observer design," *IEEE Trans. on Autom. Contr.*, vol. 39, pp. 1648–1652, 1994.

[7] K. Zhou and Z. Ren, "A new controller architecture for high performance, robust, and fault-tolerant control," *IEEE Trans. on Autom. Contr.*, vol. 46, pp. 1613–1618, 2001.

[8] C. N. Nett, C. Jacobson, and A. T. Miller, "An integrated approach to controls and diagnostics," *Proc. of ACC*, pp. 824–835, 1988.

[9] H. Niemann and J. Stoustrup, "Integration of control and fault detection: Nominal and robust design," *Proc. of the 3rd IFAC Symp. SAFEPROCESS*, vol. 1, pp. 341–346, 1997.

[10] A. Marcos and G. J. Balas, "A robust integrated controller/disgnosis aircraft application," *Int. J. of Robust and Nonlinear Contr.*, vol. 15, pp. 531–551, 2005.

[11] S. X. Ding, "Integrated design of feedback controllers and fault detectors," *Annual Reviews in Control*, vol. 33, pp. 124–135, 2009.

Part II
Fault Detection, Isolation and Estimation in Linear Dynamic Systems

Chapter 6
General Solutions of Optimal Fault Detection

In Chap. 4, Kalman filter and the unified solution or $\mathcal{H}_2$ observer-based methods have been presented as optimal solutions for fault detection in stochastic and deterministic processes, respectively. These results have been achieved on the assumption that (4.38) or more general (2.10)–(2.11) hold. In other words, the image subspace of the faults is identical with the measurement space. In real applications, the number of the faults to be detected could be smaller than the number of the sensors. In particular, by solving fault isolation problem, a bank of fault detection sub-systems are constructed, and each of them is used to detect (isolate) a special fault of considerably lower dimension. This motivates our work in this chapter to study general solutions of fault detection in dynamic systems for the case

$$\dim(f) = k_f < m = \dim(y). \tag{6.1}$$

We will propose two different types of solutions: the algebraic solution and co-inner-outer factorisation based solution. The first solution consists of straightforward matrix computations and requires less mathematical and control theoretical knowledge, while the second solution is control theoretically oriented.

6.1 Algebraic Solutions

We will deal with stochastic and deterministic processes separately, and begin with the solution for the stochastic case.

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_6

6.1.1 An Algebraic Solution for Stochastic Processes

Problem Formulation

Consider the LTI process model

$$x(k+1) = Ax(k) + Bu(k) + Ew(k) + E_f f(k), \tag{6.2}$$

$$y(k) = Cx(k) + Du(k) + v(k) + F_f f(k), \tag{6.3}$$

where $x(k) \in \mathcal{R}^n, u(k) \in \mathcal{R}^p, y(k) \in \mathcal{R}^m$ are process state, input and output vectors, respectively, and all system matrices are of appropriate dimensions and known. $f(k) \in \mathcal{R}^{k_f}$ satisfying condition (6.1) is the fault vector to be detected. $w(k) \in \mathcal{R}^{k_w}$, $v(k)$ are process and measurement noise vectors. It is assumed that they are uncorrelated with the state and input vectors, and

$$\begin{gathered} w(k) \sim \mathcal{N}\left(0, \Sigma_w\right), v(k) \sim \mathcal{N}\left(0, \Sigma_v\right), \\ \mathcal{E}\left(\begin{bmatrix} w(i) \\ v(i) \\ x(0) \end{bmatrix} \begin{bmatrix} w(j) \\ v(j) \\ x(0) \end{bmatrix}^T\right) = \begin{bmatrix} \begin{bmatrix} \Sigma_w & S_{wv} \\ S_{wv}^T & \Sigma_v \end{bmatrix} \delta_{ij} & 0 \\ 0 & \Pi_0 \end{bmatrix}. \end{gathered}$$

For the residual generation purpose, an LTI Kalman filter is applied,

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + Lr(k), \hat{x}(0) = 0, \\ r(k) &= y(k) - \hat{y}(k), \hat{y}(k) = C\hat{x}(k) + Du(k), \\ L &= \left(APC^T + ES_{wv}\right)\Sigma_r^{-1}, P = APA^T + E\Sigma_w E^T - L\Sigma_r L, \\ \Sigma_r &= CPC^T + \Sigma_v = \mathcal{E}\left(r(k)r^T(k)\right). \end{aligned}$$

The generated residual vector $r(k) \in \mathcal{R}^m$ is white and of minimum covariance matrix, and its dynamics is governed by

$$e(k+1) = (A - LC)\, e(k) + Ew(k) - Lv(k) + \left(E_f - LF_f\right) f(k), \tag{6.4}$$

$$r(k) = Ce(k) + v(k) + F_f f(k). \tag{6.5}$$

We now adopt the notations introduced in Sub-section 4.4 and write (6.4)–(6.5) into

$$
\begin{aligned}
r_s(k) &= H_o e(k-s) + H_{\bar{v},s}\bar{v}_s(k) + H_{f,s} f_s(k), \\
r_s(k) &= \begin{bmatrix} r(k-s) \\ \vdots \\ r(k) \end{bmatrix}, f_s(k) = \begin{bmatrix} f(k-s) \\ \vdots \\ f(k) \end{bmatrix}, \bar{v}_s(k) = \begin{bmatrix} \bar{v}(k-s) \\ \vdots \\ \bar{v}(k) \end{bmatrix}, \\
\bar{v}(i) &= \begin{bmatrix} w(i) \\ v(i) \end{bmatrix}, i = k-s, \cdots k, H_o = \begin{bmatrix} C \\ \vdots \\ CA_L^s \end{bmatrix}, A_L = A - LC, \\
H_{\bar{v},s} &= \begin{bmatrix} F_{\bar{v}} & 0 & & \\ CE_{\bar{v}} & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ CA_L^{s-1}E_{\bar{v}} & \cdots & CE_{\bar{v}} & F_{\bar{v}} \end{bmatrix}, E_{\bar{v}} = \begin{bmatrix} E & -L \end{bmatrix}, F_{\bar{v}} = \begin{bmatrix} 0 & I \end{bmatrix}, \\
H_{f,s} &= \begin{bmatrix} F_f & 0 & & \\ C\bar{E}_f & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ CA_L^{s-1}\bar{E}_f & \cdots & C\bar{E}_f & F_f \end{bmatrix}, \bar{E}_f = E_f - LF_f,
\end{aligned}
$$

which can be further written as

$$
\begin{aligned}
r_s(k) &= H_o A_L^{\gamma} e(k-s-\gamma) + \bar{H}_{\bar{v},s+\gamma}\bar{v}_{s+\gamma}(k) + \bar{H}_{f,s+\gamma} f_{s+\gamma}(k), \\
\Gamma_{\bar{v}} &= H_o \begin{bmatrix} A_L^{\gamma-1}E_{\bar{v}} & \cdots & A_L E_{\bar{v}} & E_{\bar{v}} \end{bmatrix}, \bar{H}_{\bar{v},s+\gamma} = \begin{bmatrix} \Gamma_{\bar{v}} & H_{\bar{v},s} \end{bmatrix}, \\
\Gamma_f &= H_o \begin{bmatrix} A_L^{\gamma-1}\bar{E}_f & \cdots & A_L\bar{E}_f & \bar{E}_f \end{bmatrix}, \bar{H}_{f,s+\gamma} = \begin{bmatrix} \Gamma_f & H_{f,s} \end{bmatrix}.
\end{aligned}
$$

Since Kalman-filter is a stable system, it holds for a large γ

$$A_L^{\gamma} \approx 0.$$

Hence, the residual vector $r_s(k)$ can be well approximated by

$$r_s(k) = \bar{H}_{\bar{v},s+\gamma}\bar{v}_{s+\gamma}(k) + \bar{H}_{f,s+\gamma} f_{s+\gamma}(k) \in \mathcal{R}^{m(s+1)}. \tag{6.6}$$

Note that

$$r_s(k) \sim \mathcal{N}\left(\mathcal{E}r_s(k), diag\left(\Sigma_r, \cdots, \Sigma_r\right)\right), \mathcal{E}r_s(k) = \begin{cases} 0, & \text{fault-free,} \\ \bar{H}_{f,s+\gamma} f_{s+\gamma}(k), & \text{faulty.} \end{cases}$$

Moreover, since

$$rank\left(\Gamma_f\right) \leq n \Longrightarrow rank\left(\bar{H}_{f,s+\gamma}\right) \leq n + (s+1)k_f,$$

for $k_f < m$ and $s \geq n$

$$n + (s+1)k_f < m(s+1) \Longrightarrow rank\left(\bar{H}_{f,s+\gamma}\right) < m(s+1). \tag{6.7}$$

As a result, our original problem of detecting faults in dynamic processes is transformed into a problem of detecting faults in a static process modelled by (6.6) and satisfying (6.7). The latter problem has been handled and solved in Sect. 3.2.

Problem Solution

We now apply the results in Sect. 3.2 to the problem solution. On the assumption

$$rank\left(\bar{H}_{f,s+\gamma}\right) = n + (s+1)k_f, \tag{6.8}$$

it holds

$$\begin{aligned}
\bar{H}^{-}_{f,s+\gamma} &= \left(\bar{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s}\bar{H}_{f,s+\gamma}\right)^{-1}\bar{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s},\\
\Sigma_{r_s} &= diag\left(\Sigma_r, \cdots, \Sigma_r\right) \in \mathcal{R}^{m(s+1)\times m(s+1)},\\
\bar{H}^{-}_{f,s+\gamma} r_s(k) &\sim \mathcal{N}\left(0, \bar{H}^{-}_{f,s+\gamma}\Sigma_{r_s}\left(\bar{H}^{-}_{f,s+\gamma}\right)^{T}\right) \text{ for } f_s(k) = 0.
\end{aligned}$$

The corresponding (optimal) test statistic and the associated threshold are

$$\begin{aligned}
J &= r_s^T(k)\left(\bar{H}^{-}_{f,s+\gamma}\right)^{T}\left(\bar{H}^{-}_{f,s+\gamma}\Sigma_{r_s}\left(\bar{H}^{-}_{f,s+\gamma}\right)^{T}\right)^{-1}\bar{H}^{-}_{f,s+\gamma} r_s(k)\\
&= r_s^T(k)\Sigma^{-1}_{r_s}\bar{H}_{f,s+\gamma}\left(\bar{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s}\bar{H}_{f,s+\gamma}\right)^{-1}\bar{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s} r_s(k)\\
&\sim \chi^2(n + (s+1)k_f), \qquad (6.9)\\
J_{th} &= \chi_\alpha, \Pr\left\{\chi^2(n + (s+1)k_f) \leq \chi_\alpha\right\} = 1 - \alpha,
\end{aligned}$$

respectively, where α is the given upper bound of the false alarm rate.

6.1.2 An Algebraic Solution for Deterministic Processes

The above solution can be extended to the deterministic case immediately. Below, we give the solution without the detailed derivation.

The process model under consideration is

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ed(k) + E_f f(k), \\ y(k) &= Cx(k) + Du(k) + Fd(k) + F_f f(k), \end{aligned}$$

where $d(k) \in \mathcal{R}^{k_d}$ represents unknown input vector and is assumed to be l_2 bounded with the known bound value

$$\|d(k)\|_2 \leq \delta_d.$$

For the residual generation purpose, an FDF is applied,

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + Lr(k), \hat{x}(0) = 0, \\ r(k) &= y(k) - \hat{y}(k), \hat{y}(k) = C\hat{x}(k) + Du(k) \end{aligned}$$

with the observer gain matrix L. The dynamics of the generated residual vector $r(k) \in \mathcal{R}^m$ is governed by

$$\begin{aligned} e(k+1) &= A_L e(k) + E_L d(k) + \bar{E}_f f(k), E_L = E - LF, \\ r(k) &= Ce(k) + Fd(k) + F_f f(k). \end{aligned}$$

It yields

$$r_s(k) = H_o e(k-s) + H_{d,s} d_s(k) + H_{f,s} f_s(k),$$

$$d_s(k) = \begin{bmatrix} d(k-s) \\ \vdots \\ d(k) \end{bmatrix}, H_{d,s} = \begin{bmatrix} F & 0 & & \\ CE_L & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ CA_L^{s-1}E_L & \cdots & CE_L & F \end{bmatrix},$$

which can be well approximated by

$$\begin{aligned} r_s(k) &= \bar{H}_{d,s+\gamma} d_{s+\gamma}(k) + \bar{H}_{f,s+\gamma} f_{s+\gamma}(k), \\ \Gamma_d &= H_o \begin{bmatrix} A_L^{\gamma-1}E_L & \cdots & A_L E_L & E_L \end{bmatrix}, \bar{H}_{d,s+\gamma} = \begin{bmatrix} \Gamma_d & H_{d,s} \end{bmatrix}. \end{aligned} \tag{6.10}$$

On the assumption

$$\begin{aligned} rank\left(\bar{H}_{f,s+\gamma}\right) &= n + (s+1)k_f < m(s+1), \\ rank\left(\bar{H}_{d,s+\gamma}\right) &= m(s+1), \end{aligned}$$

a left inverse of $\bar{H}_{f,s+\gamma}$, $\bar{H}_{f,s+\gamma}^-$, is given by

$$\bar{H}_{f,s+\gamma}^- = \left(\bar{H}_{f,s+\gamma}^T \left(\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\right)^{-1} \bar{H}_{f,s+\gamma}\right)^{-1} \bar{H}_{f,s+\gamma}^T \left(\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\right)^{-1}.$$

This allows us, using the results in Sect. 3.4, to build the residual evaluation function

$$\begin{aligned}
J &= r_s^T(k)\left(\bar{H}_{f,s+\gamma}^{-}\right)^T\left(\bar{H}_{f,s+\gamma}^{-}\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\left(\bar{H}_{f,s+\gamma}^{-}\right)^T\right)^{-1}\bar{H}_{f,s+\gamma}^{-}r_s(k)\\
&= \bar{r}_s^T(k)\left(\bar{H}_{f,s+\gamma}^T\left(\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\right)^{-1}\bar{H}_{f,s+\gamma}\right)^{-1}\bar{r}_s(k), \qquad (6.11)\\
\bar{r}_s(k) &= \bar{H}_{f,s+\gamma}^T\left(\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\right)^{-1}r_s(k),
\end{aligned}$$

with the associated threshold

$$J_{th} = \delta_d^2.$$

6.1.3 Some Remarks

In the end of this section, we would like to make remarks on the different aspects of the solutions proposed in this section.

On Real-time Implementation

In order to compute the test statistic (6.9) or the residual evaluation function (6.11) online, $s+1$ residual data, $r(i), i \in [k-s, k]$, should be first collected. There are two different ways to realise this operation:

- batch-wise data collection. That is,

$$r_s(k), r_s(k+1+s), \cdots, r_s(k+i(s+1)), \cdots, i = 0, 1, \cdots,$$

  are built for computing J defined by (6.9) or (6.11);
- moving window scheme. In this case,

$$r_s(k), r_s(k+1), \cdots, r_s(k+i), \cdots, i = 0, 1, \cdots,$$

  are built for the computation of J defined by (6.9) or (6.11).

It is evident that both test statistic (6.9) and residual evaluation function (6.11) are independent of the scheme of the data collection.

On System Structures

The key step in the proposed solutions is the computation of $\bar{H}_{f,s+\gamma}^{-}$, the left inverse of $\bar{H}_{f,s+\gamma}$. Recall that $\bar{H}_{f,s+\gamma}^{-}$ exists when condition (6.8) holds. In comparison, a realisable inverse of the transfer function (matrix) from the fault vector to the residual,

$$G_{rf}(z) = C\,(zI - A_L)^{-1}\,\bar{E}_f + F_f,$$

exists only under strict conditions like

- no transmission zero of $G_{rf}(z)$ being located outside the unit circle,
- no zero of $G_{rf}(z)$ being located at infinity.

In other words, the inverse of $G_{rf}(z)$ should be a stable and causal dynamic system. Although the so-called co-inner-outer factorisation is an alternative solution with weaker existence conditions, the needed computations are still (very) involved, as will be addressed in the next section. It follows from this comparison that the major advantages of the proposed algebraic solutions consist in

- the weak existence conditions,
- easy computation and
- no restriction on the system configuration of the solution.

The last property promises improvement of fault detection performance.

Utilisation of Information about the Fault Vector to Improve the Fault Detection Performance

Roughly speaking, the spirit of the proposed solutions is to make use of the structural information about the influence of the fault vector on the residual signal to enhance the fault detectability. The algebraic models (6.6) and (6.10) provide us with the possibility to integrate available (additional) information about the fault vector $f(k)$ into the models, in order to improve the fault detection performance. To illustrate this possibility, we give the following two examples.

Example 6.1 *Assume that $f(k)$ is an unknown constant vector denoted by f. In this case, $f_{s+\gamma}(k)$ can be written as*

$$f_{s+\gamma}(k) = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} f$$

and thus

$$\bar{H}_{f,s+\gamma}\, f_{s+\gamma}(k) = \bar{H}_{f,s+\gamma} \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} f =: \hat{H}_{f,s+\gamma}\, f,\ \hat{H}_{f,s+\gamma} \in \mathcal{R}^{m(s+1)\times k_f}.$$

As a result, we have, in case of stochastic processes,

$$
\begin{aligned}
\hat{H}^{-}_{f,s+\gamma} &= \left(\hat{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s}\hat{H}_{f,s+\gamma}\right)^{-1}\hat{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s} \in \mathcal{R}^{k_f\times m(s+1)},\\
\hat{H}^{-}_{f,s+\gamma}r_s(k) &\sim \mathcal{N}\left(0, \hat{H}^{-}_{f,s+\gamma}\Sigma_{r_s}\left(\hat{H}^{-}_{f,s+\gamma}\right)^{T}\right) \textit{ for } f=0 \Longrightarrow\\
J &= r^{T}_{s}(k)\Sigma^{-1}_{r_s}\hat{H}_{f,s+\gamma}\left(\hat{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s}\hat{H}_{f,s+\gamma}\right)^{-1}\hat{H}^{T}_{f,s+\gamma}\Sigma^{-1}_{r_s}r_s(k) \sim \chi^2(k_f),\\
J_{th} &= \chi_\alpha, \Pr\left\{\chi^2(k_f) \le \chi_\alpha\right\} = 1-\alpha,
\end{aligned}
$$

and for deterministic processes,

$$
\begin{aligned}
\hat{H}^{-}_{f,s+\gamma} &= \left(\hat{H}^{T}_{f,s+\gamma}\left(\bar{H}_{d,s+\gamma}\bar{H}^{T}_{d,s+\gamma}\right)^{-1}\hat{H}_{f,s+\gamma}\right)^{-1}\hat{H}^{T}_{f,s+\gamma}\left(\bar{H}_{d,s+\gamma}\bar{H}^{T}_{d,s+\gamma}\right)^{-1},\\
J &= \bar{r}^{T}_{s}(k)\left(\hat{H}^{T}_{f,s+\gamma}\left(\bar{H}_{d,s+\gamma}\bar{H}^{T}_{d,s+\gamma}\right)^{-1}\hat{H}_{f,s+\gamma}\right)^{-1}\bar{r}_s(k),\\
\bar{r}_s(k) &= \hat{H}^{T}_{f,s+\gamma}\left(\bar{H}_{d,s+\gamma}\bar{H}^{T}_{d,s+\gamma}\right)^{-1}r_s(k),\\
J_{th} &= \delta^2_d.
\end{aligned}
$$

Example 6.2 *Let*

$$
f(k) = \begin{bmatrix} f_1(k) \\ \vdots \\ f_{k_f}(k) \end{bmatrix}, f_{i,s+\gamma}(k) = \begin{bmatrix} f_i(k-s-\gamma) \\ \vdots \\ f_i(k) \end{bmatrix}, i = 1, \cdots, k_f.
$$

Suppose that $f_i\,(k)$ can be well approximated by

$$
f_i\,(k) = \sum_{j=0}^{s+\gamma} c_{i,j}\phi_j(k), \tag{6.12}
$$

where $\phi_j(k)$, $j = 0, 1, \cdots, s+\gamma$, are the so-called basic functions. In fact, in the context of discrete transforms of (time-domain) signals, (6.12) defines an inverse transform. There are a great number of discrete transforms. For instance, the discrete Fourier transform (DFT) is a well-known and widely applied discrete orthonormal transform, in which $c_{i,j}$, $j = 0, 1, \cdots, s+\gamma$, represent $s+\gamma+1$ frequency magnitudes. In the sequel, for the sake of better understanding, we suppose that DFT is adopted. Next, $f_{i,s+\gamma}(k)$ is written into the following compact form:

$$
\begin{aligned}
f_{i,s+\gamma}\,(k) &= \Phi C_i, i = 1, \cdots, k_f, \qquad (6.13)\\
\Phi &= \begin{bmatrix} \phi_0(k-s-\gamma) & \cdots & \phi_{s+\gamma}(k-s-\gamma) \\ \vdots & \vdots & \vdots \\ \phi_0(k) & \cdots & \phi_{s+\gamma}(k) \end{bmatrix}, C_i = \begin{bmatrix} c_{i,0} \\ \vdots \\ c_{i,s+\gamma} \end{bmatrix}.
\end{aligned}
$$

For our purpose, we transform $f_{s+\gamma}(k)$, by a regular matrix T,

$$T \in \mathcal{R}^{k_f(\gamma+s+1)\times k_f(\gamma+s+1)},$$

into

$$Tf_{s+\gamma}(k) = \begin{bmatrix} f_{1,s+\gamma}(k) \\ \vdots \\ f_{k_f,s+\gamma}(k) \end{bmatrix}$$

and re-write $f_{s+\gamma}(k)$, using (6.13), as

$$f_{s+\gamma}(k) = T^{-1} \begin{bmatrix} f_{1,s+\gamma}(k) \\ \vdots \\ f_{k_f,s+\gamma}(k) \end{bmatrix} = T^{-1} \begin{bmatrix} \Phi C_1 \\ \vdots \\ \Phi C_{k_f} \end{bmatrix}.$$

Now, it is assumed that the faults are periodic signals of q $(<< s)$ frequencies. Without loss of generality, suppose that these are the first q frequencies. Hence,

$$\begin{aligned} c_{i,j} &\neq 0, j = 0, 1, \cdots, q-1, c_{i,q+l} = 0, l = 0, 1, \cdots, s+\gamma-q, \\ \Phi C_i &= \begin{bmatrix} \phi_0(k-s-\gamma) & \cdots & \phi_{q-1}(k-s-\gamma) \\ \vdots & \vdots & \vdots \\ \phi_0(k) & \cdots & \phi_{q-1}(k) \end{bmatrix} \begin{bmatrix} c_{i,0} \\ \vdots \\ c_{i,q-1} \end{bmatrix} \\ &=: \bar{\Phi}\bar{C}_i, \bar{\Phi} \in \mathcal{R}^{(s+\gamma+1)\times q}, i = 1, \cdots, k_f, \\ f_{s+\gamma}(k) &= T^{-1} \begin{bmatrix} \bar{\Phi} & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \bar{\Phi} \end{bmatrix} \begin{bmatrix} \bar{C}_1 \\ \vdots \\ \bar{C}_{k_f} \end{bmatrix}, T^{-1} \begin{bmatrix} \bar{\Phi} & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \bar{\Phi} \end{bmatrix} \in \mathcal{R}^{k_f(s+\gamma+1)\times k_f q}. \end{aligned}$$

It leads to

$$\bar{H}_{f,s+\gamma} f_{s+\gamma}(k) = \bar{H}_{f,s+\gamma} T^{-1} \begin{bmatrix} \bar{\Phi} & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \bar{\Phi} \end{bmatrix} \begin{bmatrix} \bar{C}_1 \\ \vdots \\ \bar{C}_{k_f} \end{bmatrix} =: \hat{H}_{f,s+\gamma} \hat{f},$$

$$\hat{H}_{f,s+\gamma} \in \mathcal{R}^{m(s+1)\times k_f q}, \hat{f} = \begin{bmatrix} \bar{C}_1 \\ \vdots \\ \bar{C}_{k_f} \end{bmatrix} \in \mathcal{R}^{k_f q}.$$

Finally, we have, in case of stochastic processes,

$$\begin{aligned} J &= r_s^T(k) \Sigma_{r_s}^{-1} \hat{H}_{f,s+\gamma} \left(\hat{H}_{f,s+\gamma}^T \Sigma_{r_s}^{-1} \hat{H}_{f,s+\gamma} \right)^{-1} \hat{H}_{f,s+\gamma}^T \Sigma_{r_s}^{-1} r_s(k) \sim \chi^2(k_f q), \\ J_{th} &= \chi_\alpha, \Pr\left\{\chi^2(k_f q) \leq \chi_\alpha\right\} = 1-\alpha, \end{aligned}$$

and for deterministic processes,

$$
\begin{aligned}
J &= \bar{r}_s^T(k)\left(\hat{H}_{f,s+\gamma}^T\left(\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\right)^{-1}\hat{H}_{f,s+\gamma}\right)^{-1}\bar{r}_s(k),\\
\bar{r}_s(k) &= \hat{H}_{f,s+\gamma}^T\left(\bar{H}_{d,s+\gamma}\bar{H}_{d,s+\gamma}^T\right)^{-1}r_s(k),\\
J_{th} &= \delta_d^2.
\end{aligned}
$$

It is obvious from the both examples that the utilisation of information about the fault vector can remarkably reduce the dimension of the residual subspace, in which fault will be detected. Since such a dimension reduction does not lead to loss of information about the fault vector, the fault-to-uncertainty (noises or disturbances) ratio becomes considerably larger. In this way, the fault detectability is significantly improved.

6.2 An Observer-based Optimal Scheme

6.2.1 Problem Formulation

In this section, we propose a design scheme for an optimal observer-based fault detection, when it holds

$$k_f < m.$$

The process model under consideration is the standard LTI system,

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + Ed(k) + E_f f(k),\\
y(k) &= Cx(k) + Du(k) + Fd(k) + F_f f(k),
\end{aligned}
$$

as adopted in our previous study. The transfer function from the fault vector to the measurement is denoted by

$$G_{yf}(z) = F_f + C\left(zI - A\right)^{-1}E_f \in \mathcal{C}^{m\times k_f}.$$

Our solution consists of two main steps:

- doing a co-inner-outer factorisation of $G_{yf}(z)$ and construct a residual generator delivering a k_f-dimensional residual vector r_o,
- doing a co-inner-outer factorisation of $\hat{N}_{d,o}(z) \in \mathcal{C}^{k_f\times k_d}$, the transfer function from d to r_o, and construct the residual generator delivering residual vector r.

For the purpose of residual evaluation and threshold setting, l_2-norm of r is adopted with the corresponding threshold

$$J_{th} = \delta_d^2.$$

Although the idea behind this design scheme is similar to the algebraic solutions proposed in the last section, the system realisation is significantly different. Considering that an inverse of $G_{yf}(z)$ is a dynamic system that could be unstable and non-causal, a co-inner-outer factorisation of $G_{yf}(z)$ is to be performed.

For our purpose, we will, in the sequel, focus on solving the first design problem. It is clear that the core of the solution is the co-inner-outer factorisation of G_{yf}, which is, unfortunately, not a trivial problem.

6.2.2 A Solution

The following theorem given by Gu et al. (see the reference given in the end of this chapter) provides us with an algorithm for a (generalised) co-inner-outer factorisation of $G_{yf}(z)$.

Theorem 6.1 *Given $G_{yf}(z) \in \mathcal{C}^{m \times k_f}$, $m > k_f$ satisfying*

$$rank \begin{bmatrix} A - e^{j\theta} I & E_f \\ C & F_f \end{bmatrix} = n + k_f, \forall \theta \in [0, 2\pi],$$

and A is a Schur matrix, then there exists a co-inner-outer factorisation

$$\begin{aligned} G_{yf}(z) &= G_{f,co}(z) G_{f,ci}(z),\ G_{f,ci}(z) \in \mathcal{C}^{k_f \times k_f}, \\ G_{f,ci}(z) &= \Omega^{+} \left(F_f + C \left(zI - A_L \right)^{-1} \bar{E}_f \right), \\ G_{f,co}(z) &= \left(I + C \left(zI - A \right)^{-1} L_o \right) \Omega, \\ A_L &= A - L_o C,\ \bar{E}_f = E_f - L_o F_f, \\ \Omega \Omega^T &= \Pi = F_f F_f^T + C X_{\max} C^T, \end{aligned}$$

where the observer gain matrix L_o is given by

$$L_o = \left(A X_{\max} C^T + E_f F_f^T \right) \Pi^{+}, \tag{6.14}$$

$X_{\max}$ solves the following equation

$$X = A_L X A_L^T + \bar{E}_f \bar{E}_f^T \tag{6.15}$$

and the left inverse of $G_{f,co}(z)$ is given by

$$G_{f,co}^{-}(z) = \Omega^{+} \left(I - C \left(zI - A_L \right)^{-1} L_o \right),\ G_{f,co}^{-}(z) G_{f,co}(z) = I.$$

$\Omega \Omega^T$ is the Cholesky factorisation of Π with $\Omega \in \mathcal{R}^{m \times k_f}$, and Π^{+}, Ω^{+} are pseudo-inverse of Π, Ω with

$$\Omega^{+} \in \mathcal{R}^{k_f \times m}, rank \left(\Omega^{+} \right) = k_f.$$

Equation (6.15) has in general more than one solution for $X \geq 0$. An iterative algorithm is given by Gu et al. for finding the maximum solution $X_{\max}$. Moreover, the assumption that A is a Schur matrix loses no generality, since a stable observer can be applied before doing the co-inner-outer factorisation.

It follows from the above theorem that the dynamics of the co-inner-outer factorisation based residual generator,

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L_o\left(y(k) - \hat{y}(k)\right), \tag{6.16}$$

$$r_o(k) = \Omega^+\left(y(k) - \hat{y}(k)\right), \tag{6.17}$$

is governed by

$$\begin{aligned} e(k+1) &= A_L e(k) + \bar{E}_f f(k) + E_L d(k), E_L = E - L_o F, \\ r_o(k) &= \Omega^+\left(Ce(k) + F_f f(k) + F_d d(k)\right). \end{aligned}$$

As a result,

$$\begin{aligned} r_o(z) &= \hat{N}_{f,o}(z) f(z) + \hat{N}_{d,o}(z) d(z) \in \mathcal{C}^{k_f}, \\ \hat{N}_{f,o} &= \bar{F}_f + \bar{C}\left(zI - A_L\right)^{-1}\bar{E}_f, \left[\begin{array}{cc} \bar{F}_f & \bar{C} \end{array}\right] = \Omega^+\left[\begin{array}{cc} F_f & C \end{array}\right], \\ \hat{N}_{d,o}(z) &= \bar{F} + \bar{C}\left(zI - A_L\right)^{-1} E_L, \bar{F} = \Omega^+ F. \end{aligned}$$

Next, we design an observer-based residual generator based on $\hat{N}_{d,o}(z)$, which is equivalent with a co-inner-outer factorisation of $\hat{N}_{d,o}(z)$ and leads to the following residual generator, as described in Chap. 4,

$$\hat{x}(k+1) = (A - LC)\,\hat{x}(k) + (B - LD)\,u(k) + Ly(k), \tag{6.18}$$

$$L = L_o + L_2\Omega^+, r(k) = V_r r_o(k) = V_r\Omega^+\left(y(k) - \hat{y}(k)\right), \tag{6.19}$$

$$\begin{aligned} V_r &= \left(\bar{C}X\bar{C}^T + \bar{F}\bar{F}^T\right)^{-1/2}, L_2 = \left(A_L X\bar{C}^T + E_L\bar{F}^T\right)V_r^2, \\ & A_L X A_L^T - X + E_L E_L^T - L_2\left(\bar{C}X\bar{C}^T + \bar{F}\bar{F}^T\right)L_2^T = 0. \end{aligned}$$

The dynamics of this residual generator is governed by

$$\begin{aligned} r(z) &= \hat{N}_f(z) f(z) + \hat{N}_d(z) d(z), \\ \hat{N}_f(z) &= V_r\Omega^+\left(F_f + C\left(zI - A + LC\right)^{-1} E_{f,L}\right), \\ \hat{N}_d(z) &= V_r\Omega^+\left(F + C\left(zI - A + LC\right)^{-1} E_{d,L}\right), \\ E_{f,L} &= E_f - LF_f, E_{d,L} = E - LF, \end{aligned}$$

where $\hat{N}_d(z)$ is a co-inner. This allows us to build the evaluation function and to set the threshold as

$$J = \|r\|_2^2, J_{th} = \delta_d^2.$$

It is worth remarking that once a fault is detected, the observer-based residual generator (6.18)–(6.19) can be, for example by switching the observer gain matrix L to L_o, used for the fault estimation purpose.

6.2.3 *Discussions*

Comparing with the algebraic solutions presented in the last section, the realisation of the observer-based solution described above requires considerable mathematical and control theoretical understandings. From the fault detection point of view, the stability and causality requirements on the observer, which is in fact the inverse of a co-outer as a post-filter, imply certain system structural constraints and thus may limit the fault detection performance.

On the other hand, the algebraic solutions can also be interpreted as a residual evaluation problem of finding an optimal weighting matrix. That is, the residual evaluation function is defined by

$$J = r_s^T(k) W r_s(k), \tag{6.20}$$

and the (optimal) residual evaluation is formulated as finding a weighting matrix $W > 0$ under some given (performance) cost functions. Following this idea, the optimal fault detection problem can be formulated as

- designing an optimal observer-based residual generator (for instance, Kalman filter in the stochastic case and $\mathcal{H}_2$ observer for deterministic processes),
- determining the weighting matrix W in the evaluation function (6.20).

6.3 A Data-driven Solution

In Sect. 4.4.1, we have derived a data-driven form of SKR and, based on it, a data-driven residual generator. Notice that the dynamics of that residual generator is similar to the algebraic model form (6.6) studied in our algebraic solutions. This motivates us to give a data-driven solution.

Recall that the data-driven residual generator is constructed

$$r_s(k) = y_s(k) - K_p \begin{bmatrix} u_{s-1}(k-s-1) \\ y_{s-1}(k-s-1) \end{bmatrix} - K_{f,u} u_s(k), \tag{6.21}$$

where K_p, $K_{f,u}$ are matrices identified using the recorded process input and output data as follows:

$$\begin{bmatrix} K_p & K_{f,u} \end{bmatrix} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{+},$$

$$\begin{bmatrix} Z_p \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}.$$

Moreover, during fault-free operations, we have

$$r_s(k) = \theta, \theta \sim \mathcal{N}\left(0, L_{33}L_{33}^T\right)$$

with regular matrix $L_{33}L_{33}^T$. Suppose that we would like to detect some faults in the process and are able to model the influence of the faults on the residual vector $r_s(k)$ by means of $H_{f,s}f_s(k)$,

$$r_s(k) = H_{f,s}f_s(k) + \theta, rank\left(H_{f,s}\right) = \eta < m(s+1), \tag{6.22}$$

as shown in the following examples.

Example 6.3 *Suppose that some process faults will affect the i-th and the j-th sensors (from the m sensors, $2 < m$), and until the time instant $k - s - 1$ no fault has been detected. The influence of the faults on $y_s(k)$ is modelled by*

$$H_{f,s}f_s(k) = \begin{bmatrix} E_{ij} & & 0 \\ & \ddots & \\ 0 & & E_{ij} \end{bmatrix} \begin{bmatrix} f_i(k-s) \\ f_j(k-s) \\ \vdots \\ f_i(k) \\ f_j(k) \end{bmatrix},$$
$$E_{ij} = \begin{bmatrix} e_i & e_j \end{bmatrix} \in \mathcal{R}^{m\times 2},$$

where e_i (e_j) is a column vector with all entries equal to zero except for the i-th $(j$-th$)$ entry that is equal to one. As a result,

$$r_s(k) = H_{f,s}f_s(k) + \theta,$$
$$H_{f,s} = \begin{bmatrix} E_{ij} & & 0 \\ & \ddots & \\ 0 & & E_{ij} \end{bmatrix}, rank\left(H_{f,s}\right) = 2\left(s+1\right) < m(s+1).$$

Example 6.4 *Suppose that the number of the actuators is smaller than the number of the sensors, $p < m$, and until the time instant $k - s - 1$ no actuator faults have been detected. According to (6.21), the influence of the actuator faults represented by $f_s(k)$ on the residual vector $r_s(k)$ is modelled by*

$$\begin{aligned} r_s(k) &= H_{f,s} f_s(k) + \theta, \\ H_{f,s} &= K_{f,u} \in \mathcal{R}^{m(s+1)\times kf(s+1)}, rank\left(H_{f,s}\right) = k_f\,(s+1) < m(s+1). \end{aligned}$$

Given model (6.22), the optimal fault detection solution is obvious and summarised as follows:

$$\begin{aligned} J &= r_s^T(k)\Sigma_{r_s}^{-1} H_{f,s}\left(H_{f,s}^T \Sigma_{r_s}^{-1} H_{f,s}\right)^{-1} H_{f,s}^T \Sigma_{r_s}^{-1} r_s(k) \sim \chi^2(\eta), \\ \Sigma_{r_s}^{-1} &= \left(L_{33} L_{33}^T\right)^{-1}, \\ J_{th} &= \chi_\alpha, \Pr\left\{\chi^2(\eta) \le \chi_\alpha\right\} = 1 - \alpha. \end{aligned}$$

6.4 Notes and References

Two optimal fault detection schemes for LTI dynamic processes have been presented in this chapter for the case that

$$\dim(f) = k_f < m = \dim(y).$$

The first scheme consists of algebraic solutions including a data-driven realisation. Its core is the algebraic model of the residual dynamics (6.6). On this basis, the optimal solutions can be found by straightforward matrix computations. The achieved solutions can be interpreted as an optimal weighting of the threshold vector $r_s(k)$ as well, as given in (6.20). It is worth remarking that with the increasing dimension of $r_s(k)$, $m(s+1)$, attention should be paid to possible numerical problems. Also, computation costs could yield concern.

The second scheme is a control theoretical solution and consists of a co-inner-outer factorisation of transfer function (matrix). For our purpose, we have applied the results given by Gu et al. [1], which have been summarised as Theorem 6.1.

Recall that the optimal fault detection problem can be interpreted as an LS estimation problem that requires to inverse the mapping from the fault vector to the residual vector. For the system model expressed in terms of an algebraic relation, this can be well realised by finding a (pseudo) inverse of a matrix. For a dynamic system model, such an inverse can only be realised in form of a co-inner-outer factorisation due to the requirements on the system stability and causality. In fact, this system structural restriction limits the application of the second fault detection scheme. Also for this reason, the algebraic solutions may result in better fault detection performance.

In the end of this chapter, we would like to emphasise that the algebraic solution can be well realised in the data-driven fashion. Our discussion and examples in Sect. 6.3 have illustrated and demonstrated such a solution.

References

1. G. Gu, X.-R. Cao, and H. Badr, “Generalized LQR control and kalman filtering with relations to computations of inner-outer and spectral factorizations,” *IEEE Trans. on Autom. Control*, vol. 51, pp. 595–605, 2006.

Chapter 7
Fault Detection in Linear Time-Varying Systems

Our study on fault detection in linear discrete time-varying (LDTV) systems is highly motivated by the recent development in the fault detection research and application domains. Firstly, we see the demands for investigation on LDTV fault detection systems. It is evident that even for an LTI process the fault detection system with a finite residual evaluation horizon is time-varying. In most of studies on the LTI fault detection system design, the threshold setting is generally achieved based on the norm computation with the infinite time horizon, which may result in a conservative threshold setting. It can be observed that in practice most of evaluation schemes are realised in the discrete form and with a finite horizon.

In the application world of real-time automatic control systems, most of control systems are in their nature time-varying. For instance, SD (sampling data) and MSR (multi-sampling rate) systems are periodic and so time varying. Industrial automatic networked control systems (NCSs) with TDMA (time division multiple access) protocol can be modelled as periodic systems. The so-called event-triggered NCSs or switching systems, which are currently receiving intensive research attention, are time-varying as well.

Our objectives in this chapter are

- to derive optimal LDTV solutions for the fault detection problem defined in Chap. 2,
- to introduce various mathematical tools, including co-inner-outer factorisation (for LDTV systems), operator-aided modelling and system analysis, for dealing with fault detection issues in LDTV systems as well as their applications to the solution of the optimal FD problems,
- to reveal relations between the achieved solutions (of the defined optimal FD problem) and some optimal indices based solutions, and finally
- to demonstrate that the basic ideas for approaching fault detection in static or LTI systems can also be applied to LDTV systems.

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_7

7.1 Formulation of Fault Detection Problems

7.1.1 System Model and Assumptions

We consider LDTV systems modelled by

$$x(k+1) = A(k)x(k) + B(k)u(k) + E_d(k)d(k) + E_f(k)f(k), \tag{7.1}$$
$$y(k) = C(k)x(k) + D(k)u(k) + F_d(k)d(k) + F_f(k)f(k), \tag{7.2}$$

where $x(k) \in \mathcal{R}^n, u(k) \in \mathcal{R}^{k_u}, y(k) \in \mathcal{R}^m$ represent the state, input and output vector, respectively, $d(k) \in \mathcal{R}^{k_d}$ and $f(k) \in \mathcal{R}^{k_f}$ are unknown input vectors with $d(k)$ representing the disturbances and $f(k)$ the faults to be detected. $A(k), B(k), C(k), D(k), E_d(k), E_f(k), F_d(k)$ and $F_f(k)$ are real matrices, bounded and of appropriate dimensions, and $k_d \geq m, k_f \geq m$. It is assumed that

A1: $(C(k), A(k))$ is uniformly detectable and $(A(k), E_d(k))$ is uniformly stabilisable;
A2: $d(k), f(k)$ are $l_{2,[0,N]}$ bounded with $\|d(k)\|_{2,[0,N]} \leq \delta_{d,[0,N]}$.

7.1.2 Observer-Based FD Systems

A standard observer-based FD system is considered in our study, which consists of (i) an observer-based residual generator, (ii) a residual evaluator, and (iii) a decision logic. For the purpose of residual generation, an LDTV fault detection filter (LDTV-FDF) of the form

$$\hat{x}(k+1) = A(k)\hat{x}(k) + B(k)u(k) + L(k)\left(y(k) - \hat{y}(k)\right), \tag{7.3}$$
$$r(k) = V(k)\left(y(k) - \hat{y}(k)\right), \hat{y}(k) = C(k)\hat{x}(k) + D(k)u(k), \tag{7.4}$$

is considered. $r(k)$ is the residual vector, and the observer gain matrix $L(k)$ as well as the (static) post-filter $V(k)$ are the design parameter matrices of the FD system, respectively. The selection of $L(k)$ should ensure the exponential stability of the FDF, while $V(k) \in \mathcal{R}^{m\times m}$ is regular so that the residual subspace has the same dimension like the measurement subspace. It is straightforward that the dynamics of the residual generator is governed by

$$
\begin{aligned}
e(k+1) &= \bar{A}(k)e(k) + \bar{E}_d(k)d(k) + \bar{E}_f(k)f(k), e(k) = x(k) - \hat{x}(k), \quad (7.5)\\
r(k) &= \bar{C}(k)e(k) + \bar{F}_d(k)d(k) + \bar{F}_f(k)f(k), \quad (7.6)\\
\bar{A}(k) &= A(k) - L(k)C(k), \bar{E}_d(k) = E_d(k) - L(k)F_d(k),\\
\bar{E}_f(k) &= E_f(k) - L(k)F_f(k),\\
\bar{C}(k) &= V(k)C(k), \bar{F}_d(k) = V(k)F_d(k), \bar{F}_f(k) = V(k)F_f(k).
\end{aligned}
$$

Without loss of generality, we further assume that $\hat{x}(0) = 0$ and

A3: $\|x(0)\| = \|e(0)\| = \sqrt{e^T(0)e(0)} \leq \delta_e$.

The objective of residual evaluation is to generate a feature of the residual vector, based on which the threshold and detection logic can then be established. To this end, the $l_{2,[0,N]}$-norm of $r(k)$ is mostly used in the theoretical study,

$$J_{2,[0,N]} = \sum_{k=0}^{N} r^T(k)r(k) = \|r(k)\|_{2,[0,N]}^2 . \quad (7.7)$$

In the framework of process monitoring, there are different evaluation schemes. Analogue to them, we consider three variations of the evaluation function (7.7):

- Cumulative sum (CUSUM) scheme: the residual evaluation function is defined by

$$J_{\text{CUSUM}}(j) = \sum_{k=0}^{j} r^T(k)r(k), j = 0, 1, \cdots, N; \quad (7.8)$$

- Moving horizon (MH) scheme with the residual evaluation function defined by

$$J_{\text{MH}}(j) = \sum_{k=j}^{j+M-1} r^T(k)r(k), [j, j+M-1] \in [0, N], j = 0, 1, \cdots, N-M+1; \quad (7.9)$$

- Batch scheme: In practice, for instance in remote monitoring, data are first collected in packet and then analysed. To this end, batched residual evaluation function

$$J_{\text{batch}}(l) = \sum_{k=lM}^{(l+1)M-1} r^T(k)r(k), [lM, (l+1)M - 1] \in [0, N], l = 0, 1, \cdots , \quad (7.10)$$

  is used.

Remark 7.1 *In statistical quality control, CUSUM is the shorting for cumulative sum control chart, which is the cumulative sum of the difference between the measurement signal and the expected value, and thus different from $J_{CUSUM}(j)$ defined in (7.8).*

It is obvious that both $J_{\mathrm{CUSUM}}(j)$ and $J_{\mathrm{MH}}(j)$ are sensitive to the occurrence of a fault at each sampling time instant. In addition, it is robust against transient disturbances. Often, these residual evaluation functions can be used in a combined form.

7.1.3 Formulation of the Integrated Design of the FD Systems

The objective of our observer-based FD system design is to maximise the fault detectability in the context of Definition 2.7. Given the system model (7.1)–(7.2), the image of the disturbance vector is given by

$$\mathcal{I}_d = \left\{ y_d \left| y_d = \mathcal{M}_d(d), \, \|d\|^2_{2,[0,N]} + \|x(0)\|^2 \leq \delta^2_{d,[0,N]} + \delta^2_e \right. \right\},$$
$$\mathcal{M}_d(d) : \begin{cases} x(k+1) = A(k)x(k) + E_d(k)d(k), \\ \quad y_d(k) = C(k)e(k) + F_d(k)d(k), \end{cases}$$

and the set of undetectable faults, as given in Definition 2.6, is described by

$$\mathcal{D}_{f,undetc} = \left\{ f \left| f \in \mathcal{D}_f, \, y_f = \mathcal{M}_f(f) \in \mathcal{I}_d \right. \right\},$$
$$\mathcal{M}_f(f) : \begin{cases} x(k+1) = A(k)x(k) + E_f(k)f(k), \\ y_f(k) = C(k)x(k) + F_f(k)f(k). \end{cases}$$

Thus, our task is to (i) find the observer gain matrix $L(k)$ and the post-filter $V(k)$, (ii) set thresholds corresponding to the four residual evaluation functions given in (7.7)–(7.10) so that

$$\forall y \in \mathcal{I}_d, \, f = 0, \, J(y) - J_{th} \leq 0,$$
$$\forall f \notin \mathcal{D}_{f,undetc}, d = 0, \, J(y) - J_{th} > 0.$$

7.2 Problem Solutions

We now begin with the work on problem solutions, which will be done in a number of steps.

7.2.1 An Alternative Input-Output Model of the FD System Dynamics

For our purpose, we first re-write the dynamics of residual generator (7.5)–(7.6) and the residual evaluation functions $J_{2,[0,N]}$, $J_{\text{CUSUM}}(j)$, $J_{\text{MH}}(j)$, $J_{\text{batch}}(l)$. Let

$$\bar{r}(k_1,k_2)=\begin{bmatrix} r(k_1)\\ \vdots \\ r(k_2)\end{bmatrix}, [k_1,k_2]\in[0,N], N<\infty.$$

After a straightforward computation, we have

$$\bar{r}(k_1,k_2)=H_{\bar{d}}(k_1,k_2)\bar{d}\,(0,k_2)+H_{\bar{f}}(k_1,k_2)\bar{f}\,(0,k_2)\,, \tag{7.11}$$

$$H_{\bar{d}}(k_1,k_2)=\begin{bmatrix} H_{\bar{d}}(k_1)\\ \vdots \\ H_{\bar{d}}(k_2)\end{bmatrix}, H_{\bar{f}}(k_1,k_2)=\begin{bmatrix} H_{\bar{f}}(k_1)\\ \vdots \\ H_{\bar{f}}(k_2)\end{bmatrix}, \tag{7.12}$$

$$\begin{cases} H_{\bar{d}}(i)=\begin{bmatrix} g_e(i,0) \; g_d(i,0) \cdots g_d(i,i-1) \; g_d(i,i) \; 0 \cdots 0\end{bmatrix}, i=k_1,\cdots,k_2-1,\\ H_{\bar{d}}(k_2)=\begin{bmatrix} g_e(k_2,0) \; g_d(k_2,0) \cdots g_d(k_2,k_2-1) \; g_d(k_2,k_2)\end{bmatrix}, \end{cases}$$

$$\begin{cases} g_e(i,0)=\bar{C}(i)\Phi(i,0), i=k_1,\cdots k_2,\\ g_d(i,j)=\bar{C}(i)\Phi(i,j+1)\bar{E}_d(j), i=k_1,\cdots k_2, 0\le j<i,\\ g_d(i,i)=\bar{F}_d(i), i=k_1,\cdots k_2, \end{cases}$$

$$\begin{cases} H_{\bar{f}}(i)=\begin{bmatrix} g_f(i,0) \cdots g_f(i,i-1) \; g_f(i,i) \; 0 \cdots 0\end{bmatrix}, i=k_1,\cdots,k_2-1,\\ H_{\bar{f}}(k_2)=\begin{bmatrix} g_f(k_2,0) \cdots g_f(k_2,k_2-1) \; g_f(k_2,k_2)\end{bmatrix}, \end{cases}$$

$$g_f(i,j)=\bar{C}(i)\Phi(i,j)\bar{E}_f(j), g_f(i,i)=\bar{F}_f(i), i=k_1,\cdots,k_2, 0\le j<i,$$

$$\Phi(i,j)=\prod_{l=j}^{i-1}\bar{A}(l), \Phi(i,i)=I, i=k_1,\cdots,k_2, 0\le j<i, \tag{7.13}$$

$$\bar{d}\,(0,k_2)=\begin{bmatrix} e\,(0)\\ d\,(0)\\ \vdots \\ d\,(k_2)\end{bmatrix}, \bar{f}\,(0,k_2)=\begin{bmatrix} f\,(0)\\ \vdots \\ f\,(k_2)\end{bmatrix}.$$

Assumptions A2 and A3 can now be expressed by

$$\bar{d}^T\,(0,k_2)\,\bar{d}\,(0,k_2)\le \delta^2_{d,[0,k_2]}+\delta^2_e=:\delta^2. \tag{7.14}$$

Moreover, the previously defined four residual evaluation functions can also be respectively expressed in terms of $\bar{r}(k_1,k_2)$ with different time interval $[k_1,k_2]$:

$$J_{2,[0,N]} = \sum_{k=0}^{N} r^T(k)r(k) = \bar{r}^T(k_1,k_2)\bar{r}(k_1,k_2),\ [k_1,k_2] = [0,N],$$

$$J_{\text{CUSUM}}(j) = \sum_{k=0}^{j} r^T(k)r(k) = \bar{r}^T(k_1,k_2)\bar{r}(k_1,k_2),\ [k_1,k_2] = [0,j],$$

$$J_{\text{MH}}(j) = \sum_{k=j}^{j+M-1} r^T(k)r(k) = \bar{r}^T(k_1,k_2)\bar{r}(k_1,k_2),\ [k_1,k_2] = [j,j+M-1],$$

$$J_{\text{batch}}(l) = \sum_{k=lM}^{(l+1)M-1} r^T(k)r(k) = \bar{r}^T(k_1,k_2)\bar{r}(k_1,k_2),$$
$$[k_1,k_2] = [lM,(l+1)M-1].$$

7.2.2 The Unified Solution

We first introduce the following two lemmas, which play an important role in solving our problem as well as in the subsequent study.

Lemma 7.1 *Let*

$$L_o(k) = \left(A(k)P_o(k)C^T(k) + E_d(k)F_d^T(k)\right)V_o^2(k), \tag{7.15}$$

$$V_o(k) = \left(C(k)P_o(k)C^T(k) + F_d(k)F_d^T(k)\right)^{-1/2}, \tag{7.16}$$

where $P_o(k) > 0$ *satisfies the Riccati difference equation*

$$P_o(k+1) = \Phi_o(k+1,k)P_o(k)\Phi_o^T(k+1,k) + \bar{E}_{d,o}(k)\bar{E}_{d,o}^T(k), \tag{7.17}$$
$$\Phi_o(k+1,k) = \Phi(k+1,k)\left|_{L(k)=L_o(k)}\right.,\ \bar{E}_{d,o}(k) = \bar{E}_d(k))\left|_{L(k)=L_o(k)}\right.,$$

with $k \geq 0$, $P_o(0) = I$. *It holds*

$$H_{\bar{d},o}(0,N)H_{\bar{d},o}^T(0,N) = I,\ H_{\bar{d},o}(0,N) = H_{\bar{d}}(0,N)\left|_{L(k)=L_o(k),V(k)=V_o(k)}\right. . \tag{7.18}$$

Proof According to the definition of $H_{\bar{d},o}(i)$ given in (7.18) as well as in (7.12), it holds for $i = 0,\cdots,N$

$$H_{\bar{d},o}(i)H_{\bar{d},o}^T(i) = g_{e,o}(i,0)g_{e,o}^T(i,0) + \sum_{j=0}^{i} g_{d,o}(i,j)g_{d,o}^T(i,j)$$
$$= \bar{C}_o(i)\Gamma_o(i,0)\bar{C}_o^T(i) + \bar{F}_{d,o}(i)\bar{F}_{d,o}^T(i), \tag{7.19}$$

where

$$\begin{aligned}g_{e,o}(i,0) &= g_e(i,0)\,\Big|_{L(k)=L_o(k),V(k)=V_o(k)}\ ,\\ g_{d,o}(i,j) &= g_d(i,j)\,\Big|_{L(k)=L_o(k),V(k)=V_o(k)}\ ,0\le j\le i,\\ \bar{C}_o(i) &= V_o\,(i)\,C(i),\ \bar{F}_{d,o}(i)=V_o\,(i)\,F_d(i),\ \varGamma_o(0,0)=I,\end{aligned}$$

and for $i>0$

$$\begin{aligned}\varGamma_o(i,0) &= \varPhi_o(i,0)\varPhi_o^T(i,0)+\sum_{j=0}^{i-1}\varPhi_o(i,j+1)W(j)\varPhi_o^T(i,j+1),\\ W(j) &= \bar{E}_{d,o}(j)\bar{E}_{d,o}^T(j).\end{aligned}$$

Note that

$$\varGamma_o(i+1,0)=\varPhi_o(i+1,i)\varGamma_o(i,0)\varPhi_o^T(i+1,i)+W(i).$$

That is $\varGamma_o(i,0)$ is exactly the solution of (7.17) and thus

$$\varGamma_o(i,0)=P_o(i).$$

As a result, it follows from (7.19) and (7.16) that

$$H_{\bar{d},o}(i)H_{\bar{d},o}^T(i)=\bar{C}_o(i)P_o(i)\bar{C}_o^T(i)+\bar{F}_{d,o}(i)\bar{F}_{d,o}^T(i)=I, i=0,\cdots,N. \quad (7.20)$$

We now study

$$H_{\bar{d},o}(i)H_{\bar{d},o}^T(j)=g_{e,o}(i,0)g_{e,o}^T(j,0)+\sum_{l=0}^{i}g_{d,o}(i,l)g_{d,o}^T(j,l),\ j=1,\cdots,N,$$

for $0\le i<j$. Let $\bar{A}_o(i)=A(i)-L_o(i)C(i)$, it turns out

$$\begin{aligned}g_{e,o}(i,0)g_{e,o}^T(j,0) &= \bar{C}_o(i)\varPhi_o(i,0)\varPhi_o^T(i,0)\bar{A}_o^T(i)\varPhi_o^T(j,i+1)\bar{C}_o^T(j),\\ g_{d,o}(i,l)g_{d,o}^T(j,l) &= \bar{C}_o(i)\varPhi_o(i,l+1)W(l)\varPhi_o^T(i,l+1)\bar{A}_o^T(i)\varPhi_o^T(j,i+1)\bar{C}_o^T(j),\\ g_{d,o}(i,i)g_{d,o}^T(j,i) &= V_o\,(i)\,F_d(i)\bar{E}_{d,o}^T(i)\varPhi_o^T(j,i+1)\bar{C}_o^T(j),\end{aligned}$$

for $l=0,\cdots,i$. It yields

$$\begin{aligned}
H_{\bar{d},o}(i)H_{\bar{d},o}^T(j) &= V_o\,(i)\,\Psi(i)\Phi^T(j,i+1)\bar{C}_o^T\,(j)\,,\\
\Psi(i) &= C(i)\left(\Phi_o(i,0)\Phi_o^T(i,0)+\sum_{l=N}^{i-1}\Phi_o(i,l+1)W(l)\Phi_o^T(i,l+1)\right)\bar{A}_o^T(i)\\
&\quad + F_d(i)\bar{E}_{d,o}^T(j).
\end{aligned}$$

Remember that

$$\Phi_o(i,0)\Phi_o^T(i,0)+\sum_{l=N}^{i-1}\Phi_o(i,l+1)W(l)\Phi_o^T(i,l+1) = P_o(i)$$

and moreover $\forall k$

$$\begin{aligned}
&L_o(k)\left(C(k)P_o(k)C^T(k)+F_d(k)F_d^T(k)\right) = A(k)P_o(k)C^T(k)+E_d(k)F_d^T(k)\\
&\Longrightarrow \bar{A}_o(k)P_o(k)C^T(k)+\bar{E}_{d,o}(k)F_d^T(k)=0. \qquad (7.21)
\end{aligned}$$

Thus, we have for $j=1,\cdots,N, 0\le i<j$,

$$\begin{aligned}
\Psi^T(i) &= \bar{A}_o(i)P_o(i)C^T(i)+\bar{E}_{d,o}(i)F_d^T(i)=0\\
&\Longrightarrow H_{\bar{d},o}(i)H_{\bar{d},o}^T(j) = H_{\bar{d},o}^T(j)H_{\bar{d},o}(i)=0.
\end{aligned}$$

As a result of the above equation and (7.20), (7.18) is finally proved.

Remark 7.2 *For the practical implementation, the stability of the LDTV-FDF is required. It is well-known that under Assumption A1 the FDF with*

$$\Phi_o(k+1,k) = A(k)-L_o(k)C(k)$$

is exponentially stable.

Along the lines of the above proof, a general form of (7.18) in Lemma 7.1, which is given below, can be easily proved:

$$\begin{aligned}
&H_{\bar{d},o}(k_1,k_2)H_{\bar{d},o}^T(k_1,k_2) = I, \forall\,[k_1,k_2]\in[0,N], \qquad (7.22)\\
&H_{\bar{d},o}(k_1,k_2) = H_{\bar{d}}(k_1,k_2)\,\big|_{L(k)=L_o(k),V(k)=V_o(k)}\ .
\end{aligned}$$

Lemma 7.2 *Given $V_o(k), L_o(k), k\in[0,k_1-1]$, as defined in (7.15)–(7.16) in Lemma 7.1, then for any (regular) $V(k)$ and $L(k)$ with $k\in[k_1,k_2]$, it holds*

$$\begin{aligned}
H_{\bar{d}}(k_1,k_2) &= Q\,(k_1,k_2)\,H_{\bar{d},o}(k_1,k_2), && (7.23)\\
H_{\bar{f}}(k_1,k_2) &= Q\,(k_1,k_2)\,H_{\bar{f},o}(k_1,k_2), && (7.24)\\
H_{\bar{f},o}(k_1,k_2) &= H_{\bar{f}}(k_1,k_2)\left.\right|_{L(k)=L_o(k),V(k)=V_o(k)}\ ,\\
Q\,(k_1,k_2) &= \begin{bmatrix} \bar{V}(k_1) & 0 & \cdots & & 0\\ \Upsilon(k_1+1,k_1) & \bar{V}(k_1+1) & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ \Upsilon(k_2-1,k_1) & \cdots & \Upsilon(k_2-1,k_2-2) & \bar{V}(k_2-1) & 0\\ \Upsilon(k_2,k_1) & \cdots & \Upsilon(k_2,k_2-2) & \Upsilon(k_2,k_2-1) & \bar{V}(k_2) \end{bmatrix}\\
\bar{V}(k) &= V(k)V_o^{-1}(k),\ \Upsilon(k,j)=\bar{C}(k)\Phi(k,j+1)\Delta L(j),\\
\Delta L(j) &= (L_o(j)-L(j))\,V_o^{-1}(j),\ k=k_1,\cdots,k_2,k_1\le j<k.
\end{aligned}$$

Proof We prove (7.23). Given $V_o(0)$, $L_o(0)$, ..., $V_o(k_1-1)$, $L_o(k_1-1)$, it is obvious that

$$g_e(k_1,0)=\bar{C}(k_1)\Phi_o(k_1,0)=\bar{V}(k_1)g_{e,o}(k_1,0).$$

For $k=k_1+1,\cdots,k_2$,

$$\begin{aligned}
g_e(k,0) &= \bar{C}(k)\Phi(k,0)\\
&= \bar{C}(k)\Phi(k,k_1+1)\left(\Phi_o(k_1+1,k_1)+\Delta L(k_1)V_o(k_1)C(k_1)\right)\Phi_o(k_1,0)\\
&= \bar{C}(k)\Phi(k,k_1+1)\Phi_o(k_1+1,0)+\bar{C}(k)\Phi(k,k_1+1)\Delta L(k_1)g_{e,o}(k_1,0)\\
&= \cdots\\
&= \bar{C}(k)\sum_{l=k_1}^{k-1}\Phi(k,l+1)\Delta L(l)V_o(l)C(l)\Phi_o(l,0)+\bar{C}(k)\Phi_o(k,0)\\
&= \sum_{l=k_1}^{k-1}\Upsilon(k,l)g_{e,o}(l,0)+\bar{V}(k)g_{e,o}(k,0).
\end{aligned}$$

Similarly, for $k=k_1+1,\cdots,k_2$, $0\le j<k_1$,

$$\begin{aligned}
g_d(k,j) &= \bar{C}(k)\Phi(k,k_1)\Phi_o(k_1,j+1)\bar{E}_{d,o}(j)\\
&= \bar{C}(k)\Phi_o(k,j+1)\bar{E}_{d,o}(j)+\bar{C}(k)\sum_{l=k_1}^{k-1}\Phi(k,l+1)\Delta L(l)\bar{C}_o(l)\Phi_o(l,j+1)\bar{E}_{d,o}(j)\\
&= \sum_{l=k_1}^{k-1}\Upsilon(k,l)g_{d,o}(l,j)+\bar{V}(k)g_{d,o}(k,j),
\end{aligned}$$

and for $k_1\le j<k$

$$\begin{aligned} g_d(k,j) &= \bar{C}(k)\Phi(k,j+1)\bar{E}_d(j) \\ &= \bar{C}(k)\Phi(k,j+1)(\bar{E}_{d,o}(j) + \Delta L(j)V_o(j)F_d(j)) \\ &= \cdots \\ &= \bar{C}(k)\sum_{l=j+1}^{k-1}\Phi(k,l+1)\Delta L(l)\bar{C}_o(l)\Phi_o(l,j+1)\bar{E}_{d,o}(j) \\ &\quad + \bar{C}(k)\Phi(k,j+1)\Delta L(j)\bar{F}_{d,o}(j) \\ &= \sum_{l=j}^{k-1}\Upsilon(k,l)g_{d,o}(l,j), \end{aligned}$$

and finally for $j = k$,

$$g_d(j,j) = V(j)\,F_d(j) = V(j)\,V_o^{-1}(j)V_o(j)F_d(j) = \bar{V}(j)\,g_{d,o}(j,j).$$

Remember that

$$\begin{aligned} H_{\bar{d},o}(k_1,k_2) &= \begin{bmatrix} H_{\bar{d},o}(k_1) \\ \vdots \\ H_{\bar{d},o}(k_2) \end{bmatrix} \\ &= \begin{bmatrix} g_{e,o}(k_1,0) & g_{d,o}(k_1,0) & \cdots & g_{d,o}(k_1,k_1) & 0 & \cdots \\ \vdots & \vdots & & & \ddots & \ddots \\ g_{e,o}(k_2,0) & g_{d,o}(k_2,0) & & \cdots & & g_{d,o}(k_2,k_2) \end{bmatrix}. \end{aligned}$$

It directly follows from the above results that

$$H_{\bar{d}}(k_1,k_2) = Q(k_1,k_2)\,H_{\bar{d},o}(k_1,k_2)$$

Note that $Q(k_1,k_2)$ is independent of $\bar{E}_d$, F_d and $g_f(k,j)$ is similar with $g_d(k,j)$ by replacing $\bar{E}_d$, F_d with $\bar{E}_f$, F_f. The proof of (7.24) is analogue to the one of (7.23). As a result, (7.23) and (7.24) are proved.

As a special case of Lemma 7.2, for $k_1 = 0, k_2 \in (0,N]$, it holds

$$H_{\bar{d}}(0,k_2) = Q(0,k_2)\,H_{\bar{d},o}(0,k_2),\; H_{\bar{f}}(0,k_2) = Q(0,k_2)\,H_{\bar{f},o}(0,k_2). \tag{7.25}$$

Noting that $Q(0,k_2)$ is invertible and

$$H_{\bar{d},o}(0,k_2) = Q^{-1}(0,k_2)\,H_{\bar{d}}(0,k_2), \tag{7.26}$$

$Q^{-1}(0,k_2)$ has the identical structure like $Q(0,k_2)$ and is formed by changing the positions of $V_o(k)$ and $V(k)$ and the positions of $L_o(k)$ and $L(k)$ in the related equations as follows:

$$Q^{-1}(0,k_2)=\begin{bmatrix} \bar{V}(0) & 0 & \cdots & & 0 \\ \Upsilon(1,0) & \bar{V}(1) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \Upsilon(k_2-1,0) & \cdots & \Upsilon(k_2-1,k_2-2) & \bar{V}(k_2-1) & 0 \\ \Upsilon(k_2,0) & \cdots & \Upsilon(k_2,k_2-2) & \Upsilon(k_2,k_2-1) & \bar{V}(k_2) \end{bmatrix},$$

$$\bar{V}(k)=V_o(k)V^{-1}(k), \Upsilon(k,j)=\bar{C}_o(k)\Phi_o(k,j+1)\Delta L(j),$$
$$\Delta L(j)=\left(L(j)-L_o(j)\right)V^{-1}(j), k=0,\cdots,k_2, 0\leq j<k.$$

Moreover, let for $k\in[0,N]$

$$V(k)=I, L(k)=0,$$
$$y_d(0,k)=\begin{bmatrix} y(0) \\ \vdots \\ y(k) \end{bmatrix} \text{ for } u(k)=0, f(k)=0, \tag{7.27}$$
$$y_f(0,k)=\begin{bmatrix} y(0) \\ \vdots \\ y(k) \end{bmatrix} \text{ for } x(0)=0, u(k)=0, d(k)=0. \tag{7.28}$$

It is obvious that the influences of $x(0), d(k), f(k)$ on the system output $y(k)$ (that is in the open-loop configuration) can be equivalently described by

$$\bar{r}(0,k)=y_d(0,k)+y_f(0,k)=H_{\bar{d}}(0,k)\bar{d}(0,k)+H_{\bar{f}}(0,k)\bar{f}(0,k)$$
$$=Q(0,k)H_{\bar{d},o}(0,k)\bar{d}(0,k)+Q(0,k)H_{\bar{f},o}(0,k)\bar{f}(0,k)$$

with $V(k)=I, L(k)=0$. In this way, we have proved the following lemma.

Lemma 7.3 *Given LDTV system model (7.1)–(7.2), then the influences of* $x(0), d(k)$ *on the system output* $y(k)$, $y_d(0,k)$ *for* $k\in[0,N]$, *can be factorised by*

$$y_d(0,k)=H_{\bar{d}}(0,k)\bar{d}(0,k), H_{\bar{d}}(0,k)=Q(0,k)H_{\bar{d},o}(0,k), \tag{7.29}$$

where $y_d(0,k)$ *is defined by (7.27).*

To be consistent with the notations introduced in Chapter 2 and Sect. 7.1, we denote

$$y_d=y_d(0,k), y_f=y_f(0,k),$$
$$\mathcal{M}_d(d)=H_{\bar{d}}(0,k)\bar{d}(0,k), \mathcal{M}_f(f)=H_{\bar{f}}(0,k)\bar{f}(0,k),$$
$$\mathcal{I}_d=\left\{y_d(0,k)\left|y_d(0,k)=H_{\bar{d}}(0,k)\bar{d}(0,k), \left\|\bar{d}(0,k)\right\|^2\leq\delta^2_{d,[0,k]}+\delta^2_e\right.\right\},$$
$$\mathcal{D}_{f,undetc}=\left\{\bar{f}(0,k)\left|y_f(0,k)=H_{\bar{f}}(0,k)\bar{f}(0,k)\in\mathcal{I}_d\right.\right\}.$$

We are now in a position to present the first solution to our FD problem formulated in Sect. 7.1.

Theorem 7.1 *Given the LDTV system model (7.1)–(7.2), the FDF (7.3)–(7.4) and the residual evaluation functions (7.7) as well as (7.8), then $V_o(k)$, $L_o(k)$ given in (7.15)–(7.16) in Lemma 7.1 and*

$$J_{th,2,[0,N]} = \delta^2_{d,[0,N]} + \delta^2_e, \tag{7.30}$$

$$J_{th,2,CUSUM} = \delta^2_{d,[0,N]} + \delta^2_e \tag{7.31}$$

solve the optimal FD problem.

Proof Viewing CUSUM evaluation function (7.8) as $J_{2,[0,j]}$ evaluation function with a varying j, the proof for the optimal FD with CUSUM evaluation function can be easily achieved by extending the proof for the FD with evaluation function (7.7). Therefore, we focus on the latter case. Recall that

$$\begin{aligned} \mathcal{M}_d(d) &= H_{\bar{d}}(0,N)\bar{d}\,(0,N) = Q\,(0,N)\,H_{\bar{d},o}(0,N)\bar{d}\,(0,N) \\ &\Longleftrightarrow H_{\bar{d},o}(0,N)\bar{d}\,(0,N) = Q^{-1}\,(0,N)\,H_{\bar{d}}(0,N)\bar{d}\,(0,N)\,. \end{aligned}$$

According to Theorem 2.1, we only need to prove

$$\mathcal{M}^-_{\bar{d}} = Q^{-1}\,(0,N)\,.$$

Since $Q^{-1}\,(0,N)$ is invertible and

$$H_{\bar{d},o}(0,N)H^T_{\bar{d},o}(0,N) = I,$$

it is obvious that

- $\forall \bar{d}\,(0,N)$,

$$\begin{aligned} \|\bar{r}(0,N)\|^2 = \|r(k)\|^2_{2,[0,N]} &= \left\|H_{\bar{d},o}(0,N)\bar{d}\,(0,N)\right\|^2 \\ &= \left\|\mathcal{M}^-_{\bar{d}} \circ \mathcal{M}_d(d)\right\|^2 \le \delta^2_{d,[0,N]} + \delta^2_e \end{aligned}$$

- $\forall \bar{f}\,(0,k) \in \mathcal{D}_{f,undetc}$, $\exists \bar{d}\,(0,k)$ so that

$$\begin{aligned} &H_{\bar{f}}(0,k)\bar{f}\,(0,k) = H_{\bar{d},o}(0,k)\bar{d}\,(0,k) \Longrightarrow \\ &\left\|H_{\bar{f}}(0,k)\bar{f}\,(0,k)\right\|^2 = \left\|\bar{d}\,(0,k)\right\|^2. \end{aligned}$$

As a result, it follows from Theorem 2.1 that $Q^{-1}\,(0,N)$ with $V_o(k)$, $L_o(k)$ defined in (7.15)–(7.16) in Lemma 7.1 and threshold $J_{th,2,[0,N]}$ given in (7.30) solve the optimal FD problem. The theorem is proved.

Noting that the proof given above holds for any integer N, an extension of this proof to the optimal FD with CUSUM evaluation function is straightforward, since for each j the given matrices $V_o(k)$, $L_o(k)$ as well as $J_{th,CUSUM}$ are optimal.

The optimal solutions corresponding to the other two evaluation functions are summarised in the following theorem, whose proof is similar to the one of the above theorem.

Theorem 7.2 *Given the LDTV system model (7.1)–(7.2), the FDF (7.3)–(7.4), and the residual evaluation functions (7.9) and (7.10), then $V_o(k)$, $L_o(k)$ given in (7.15)–(7.16) in Lemma 7.1 and*

$$J_{th,MH} = \delta^2_{d,[0,N]} + \delta^2_e, \tag{7.32}$$

$$J_{th,batch} = \delta^2_{d,[0,N]} + \delta^2_e \tag{7.33}$$

solve the optimal FD problem.

Proof We begin with the residual evaluation functions (7.9) and (7.10) for $j = l = 0$, which are equivalent to the FD problem in the time interval $[0, M-1]$ with residual evaluation function

$$J = \|r(k)\|^2_{2,[0,M-1]}\,.$$

Thus, according to Theorem 7.1, the optimal solution, for both cases, is given by $V_o(k)$, $L_o(k)$ in (7.15)–(7.16) and the threshold settings (7.32) and (7.33), respectively. Next, for $j = l = 1$, the FD problems are equivalent to the FD problem in the time intervals $[1, M]$ as well as $[M, 2M-1]$ with residual evaluation functions

$$J = \|r(k)\|^2_{2,[1,M]} \text{ as well as } J = \|r(k)\|^2_{2,[M,2M-1]}\,.$$

Denote the influences of $x(0), d(k), f(k)$ on the system output $y(k)$ in the time intervals $[1, M]$ as well as $[M, 2M-1]$ by

$$\begin{aligned} y_d(k_1,k_2) + y_f(k_1,k_2) &= H_{\bar{d}}(k_1,k_2)\bar{d}\,(0,k_2) + H_{\bar{f}}(k_1,k_2)\bar{f}\,(0,k_2)\,, \\ &\quad [k_1,k_2] = [1,M] \text{ as well as } [k_1,k_2] = [M,2M-1], \end{aligned}$$

which is equivalent to $\bar{r}(k_1,k_2)$ with $V(k) = I, L(k) = 0$. It follows from Lemma 7.2 that

$$\begin{aligned} & H_{\bar{d}}(k_1,k_2)\bar{d}\,(0,k_2) + H_{\bar{f}}(k_1,k_2)\bar{f}\,(0,k_2) \\ & = Q(k_1,k_2)H_{\bar{d},o}(k_1,k_2)\bar{d}\,(0,k_2) + Q(k_1,k_2)H_{\bar{f},o}(k_1,k_2)\bar{f}\,(0,k_2) \Longleftrightarrow \\ & Q^{-1}(k_1,k_2)\left(H_{\bar{d}}(k_1,k_2)\bar{d}\,(0,k_2) + H_{\bar{f}}(k_1,k_2)\bar{f}\,(0,k_2)\right) \\ & = H_{\bar{d},o}(k_1,k_2)\bar{d}\,(0,k_2) + H_{\bar{f},o}(k_1,k_2)\bar{f}\,(0,k_2)\,. \end{aligned}$$

Re-write the above equation in the form of

$$H_{\bar{d}}(k_1,k_2)\bar{d}\,(0,k_2) + H_{\bar{f}}(k_1,k_2)\bar{f}\,(0,k_2) = \mathcal{M}_d(d) + \mathcal{M}_f(f)$$

and recall (7.22),

$$H_{\bar{d},o}(k_1,k_2)H_{\bar{d},o}^T(k_1,k_2) = I.$$

As a result, along the lines of the proof of Theorem 7.1, it is obvious that

$$\mathcal{M}_d^- = Q^{-1}(k_1,k_2)$$

delivers the optimal solution. Notice that the above results hold for any time interval $[k_1,k_2]$. That is, they are also true for

$$[k_1,k_2] = [j, j+M-1] \text{ as well as } [k_1,k_2] = [lM,(l+1)M-1].$$

We have finally proved that $V_o(k)$, $L_o(k)$ given in (7.15)–(7.16) and the threshold settings (7.32) and (7.33) solve the optimal FD problem.

With Theorems 7.1 and 7.2, we have now the complete solution for the optimal FD problem formulated in Definition 2.7. It is remarkable that for all four variations of the residual evaluation functions we have the identical solution. It is also worth noting that the unified solution presented in Sect. 4.3 for LTI systems is a special case of the solution given above. With this background as well as our subsequent discussions, we continue to use "unified solution" to denote the solution given in Theorems 7.1 and 7.2.

In the subsequent sections, we shall discuss about the solution from different aspects and using various system modelling and analysis techniques, which will result in diverse interesting interpretations.

7.3 Algebraic I/O-Model, Co-inner-outer Factorisation and Unified Solution

In the previous section, we have found the optimal fault detection solution using an input-output model of the observer-based FD system dynamics. In this section, we address the same fault detection problem on the basis of the algebraic input-output model adopted in our investigation on parity space approach and data-driven fault detection in Chap. 4.

7.3.1 The Algebraic Input-Output Model for LDTV Systems

Given LDTV system model (7.1)–(7.2), its algebraic input-output model is described by

$$y_k(k) = H_{o,k}x(0) + H_{u,k}u_k(k) + H_{d,k}d_k(k) + H_{f,k}f_k(k), \tag{7.34}$$

$$y_k(k) = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k) \end{bmatrix}, u_k(k) = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(k) \end{bmatrix}, d_k(k) = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(k) \end{bmatrix}, f_k(k) = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(k) \end{bmatrix},$$

$$H_{o,k} = \begin{bmatrix} C(0)\Psi(0,0) \\ C(1)\Psi(1,0) \\ \vdots \\ C(k)\Psi(k,0) \end{bmatrix}, \Psi(i,j) = \prod_{l=j}^{i-1} A(l), \Psi(i,i) = I,$$

$$H_{u,k} = \begin{bmatrix} D(0) & 0 & \cdots & 0 \\ C(1)\Psi(0,0)B(0) & D(1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C(k)\Psi(k-1,0)B(0) & \cdots & C(k)\Psi(0,0)B(k-1) & D(k) \end{bmatrix},$$

and $H_{d,k}$, $H_{f,k}$ have the identical structure like $H_{u,k}$ and are built by substituting $D(k)$, $B(k)$ in terms of $F_d(k)$, $F_d(k)$ and $E_d(k)$, $E_f(k)$, respectively.

Recalling the definitions of matrices $H_{\bar{d}}(k_1, k_2)$, $H_{\bar{f}}(k_1, k_2)$ as well as vectors $\bar{d}\,(0, k_2)$, $\bar{f}\,(0, k_2)$ given in (7.11)–(7.13), the algebraic input-output model (7.34) can be re-written as

$$y_k(k) = H_{u,k}u_k(k) + H_{\bar{d}}(0,k)\bar{d}\,(0,k) + H_{\bar{f}}(0,k)\bar{f}\,(0,k)\,, \tag{7.35}$$

where $V(k)$, $L(k)$ in $H_{\bar{d}}(0,k)$, $H_{\bar{f}}(0,k)$ are set to be

$$V(k) = I, L(k) = 0. \tag{7.36}$$

In our subsequent discussion, matrices $H_{\bar{d}}(0,k)$, $H_{\bar{f}}(0,k)$ are considered on the assumption of (7.36).

7.3.2 Co-inner-outer Factorisation and the Solution

It follows from Lemmas 7.1–7.3 that

$$\begin{bmatrix} H_{o,k} & H_{d,k} \end{bmatrix} = H_{\bar{d}}(0,k) = Q\,(0,k)\,H_{\bar{d},o}(0,k), \tag{7.37}$$

$$H_{\bar{d},o}(0,k)H_{\bar{d},o}^T(0,k) = I. \tag{7.38}$$

Moreover, $Q\,(0,k)$ is invertible and given by

$$Q\,(0,k) = \begin{bmatrix} V_o(0) & 0 & \cdots & & 0 \\ \Upsilon(1,0) & V_o(1) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \Upsilon(k-1,0) & \cdots & \Upsilon(k-1,k-2) & V_o(k-1) & 0 \\ \Upsilon(k,0) & \cdots & \Upsilon(k,k-2) & \Upsilon(k,k-1) & V_o(k) \end{bmatrix},$$
$$\Upsilon(i,j) = -\bar{C}_o(i)\Phi_o(i,j+1)L_o(j), i = 1,\cdots,k, 0 \le j < i.$$

We call (7.37)–(7.38) a co-inner-outer factorisation of $\begin{bmatrix} H_{o,k} & H_{d,k} \end{bmatrix}$ with $Q\,(0,k)$ as co-outer and $H_{\bar{d},o}(0,k)$ as co-inner. Analogue to the relation between the co-inner-outer factorisation of LTI systems and unified solution presented in Sect. 4.3, we see, from the co-inner-outer factorisation of $\begin{bmatrix} H_{o,k} & H_{d,k} \end{bmatrix}$, another aspect of the optimal residual generator (7.3)–(7.4) with $L(k) = L_o(k)$, $V(k) = V_o(k)$. To this end, re-form the input-output model (7.34) into

$$\begin{aligned} y_k(k) - H_{u,k}u_k(k) &= H_{o,k}x(0) + H_{d,k}d_k(k) + H_{f,k}f_k(k) \\ &= H_{\bar{d}}(0,k)\bar{d}\,(0,k) + H_{\bar{f}}(0,k)\bar{f}\,(0,k)\,. \end{aligned}$$

By means of the co-inner-outer factorisation (7.37)–(7.38), it turns out

$$\begin{aligned} &y_k(k) - H_{u,k}u_k(k) = Q\,(0,k)\left(H_{\bar{d},o}(0,k)\bar{d}\,(0,k) + H_{\bar{f},o}(0,k)\bar{f}\,(0,k)\right) \Longrightarrow \\ &Q^{-1}\,(0,k)\left(y_k(k) - H_{u,k}u_k(k)\right) = H_{\bar{d},o}(0,k)\bar{d}\,(0,k) + H_{\bar{f},o}(0,k)\bar{f}\,(0,k)\,. \end{aligned}$$

It is clear that

$$Q^{-1}\,(0,k)\left(y_k(k) - H_{u,k}u_k(k)\right) = \bar{r}(0,k),$$

and finally it holds, in the fault-free case,

$$J_{2,[0,k]} = \sum_{i=0}^{k} r^T(i)r(i) = \|\bar{r}(0,k)\|^2 \le \left\|\bar{d}\,(0,k)\right\|^2 \le \delta^2_{d,[0,k]} + \delta^2_e.$$

Consider the dual form of

$$\bar{r}(0,k) = H_{\bar{d},o}(0,k)\bar{d}\,(0,k)\,,$$

the residual vector in the fault-free case, which can be written as

$$\hat{\bar{d}}\,(0,k) = H^T_{\bar{d}}(0,k)\bar{r}(0,k), \tag{7.39}$$

and interpreted as a residual-based estimation for $\bar{d}\,(0,k)$. Recall that

$$H_{\bar{d},o}(0,k)H^T_{\bar{d},o}(0,k) = I.$$

Hence,

$$\|r(i)\|^2_{2,[0,k]} = \|\bar{r}(0,k)\|^2 = \left\|\hat{\bar{d}}\,(0,k)\right\|^2. \tag{7.40}$$

That means, the unified solution can be equivalently viewed as a residual-driven estimation of the disturbance with the property that the $l_{2,[0,k]}$ norm of the residual vector equals to the one of the estimated disturbance vector.

It is of interest to notice that the state space realisation of the algebraic input-output model (7.39) is given by

$$\bar{e}(i) = \bar{A}^T_o(i)\bar{e}(i+1) + \bar{C}^T_o(i)\bar{r}(i+1), \tag{7.41}$$

$$\hat{d}(i+1) = \bar{E}^T_{d,o}(i)\bar{e}(i+1) + \bar{F}^T_{d,o}(i)\bar{r}(i+1) \tag{7.42}$$

with $\bar{e}(k+1) = 0, i \in [0,k]$, and

$$\hat{\bar{d}}\,(0,k) = \begin{bmatrix} \bar{e}(0) \\ \hat{d}(1) \\ \vdots \\ \hat{d}(k+1) \end{bmatrix}, \bar{r}(0,k) = \begin{bmatrix} r(0) \\ \vdots \\ r(k) \end{bmatrix} = \begin{bmatrix} \bar{r}(1) \\ \vdots \\ \bar{r}(k+1) \end{bmatrix}. \tag{7.43}$$

It is well-known in control theory that the system representation (7.41)–(7.42) is the dual form of the system model

$$e(i+1) = \bar{A}_o(i)e(i) + \bar{E}_{d,o}(i)d(i), r(i) = \bar{C}_o(i)e(i) + \bar{F}_{d,o}(i)d(i). \tag{7.44}$$

7.4 Co-inner-outer, Lossless and Unified Solution

In control theory, an inner system is lossless with respect to the so-called l_2-gain supply rate. Generally speaking, a system is lossless with respect to the l_2-gain supply rate, when the energy change stored in the system is equal to the difference between the system input and output energy. In some literatures, an inner system is also defined in the context of this energy balance relation. As a dual form of the inner system with (energy) lossless behaviour, we introduce first the following definition of a co-inner system in the context of lossless information transform.

Definition 7.1 *Given system*

$$x(k+1) = A(k)x(k) + E(k)d(k), r(k) = C(k)x(k) + D(k)d(k) \tag{7.45}$$

and its dual form

$$\bar{x}(k) = A^T(k)\bar{x}(k+1) + C^T(k)\bar{r}(k+1), \tag{7.46}$$
$$\bar{d}(k+1) = E^T(k)\bar{x}(k+1) + D^T(k)\bar{r}(k+1), \tag{7.47}$$

system (7.45) is called co-inner if there exists a function $V(\bar{x}(k)) \geq 0$, $V(0) = 0$ *such that*

$$V(\bar{x}(N+1)) - V(\bar{x}(0)) = \left\|\bar{d}(k+1)\right\|_{2,[0,N]}^2 - \|\bar{r}(k+1)\|_{2,[0,N]}^2. \tag{7.48}$$

In the framework of process monitoring and diagnosis, $V(\bar{x}(k))$ and $\left\|\bar{d}(k+1)\right\|_{2,[0,N]}^2$ can be viewed as uncertainties due to the (not measurable) system state variables and unknown input $\bar{d}(k+1)$, while $\|\bar{r}(k+1)\|_{2,[0,N]}^2$ represents the information amount about the uncertainties, which can be extracted from the system measurement. In this context, we call system (7.45) lossless with respect to information transform rate

$$s(\bar{d}, \bar{r}) = \bar{d}^T(k+1)\bar{d}(k+1) - \bar{r}^T(k+1)\bar{r}(k+1).$$

Theorem 7.3 *Consider residual dynamics in the fault-free case,*

$$e(k+1) = \bar{A}_o(k)e(k) + \bar{E}_{d,o}(k)d(k), \tag{7.49}$$
$$r(k) = \bar{C}_o(k)e(k) + \bar{F}_{d,o}(k)d(k), \tag{7.50}$$

and its dual form

$$\bar{e}(k) = \bar{A}_o^T(k)\bar{e}(k+1) + \bar{C}_o^T(k)\bar{r}(k+1),$$
$$\hat{d}(k+1) = \bar{E}_{d,o}^T(k)\bar{e}(k+1) + \bar{F}_{d,o}^T(k)\bar{r}(k+1),$$

for $k \in [0, N]$ *and* $\bar{e}(N+1) = 0$, *where*

$$\bar{A}_o(k) = A(k) - L_o(k)C(k), \bar{C}_o(k) = V_o(k)C(k),$$
$$\bar{E}_{d,o}(i) = E_d(k) - L_o(k)F_d(k), \bar{F}_{d,o}(k) = V_o(k)F_d(k)$$

with $L_o(k)$, $V_o(k)$ *defined in Lemma 7.1. Then, system (7.49) is co-inner and lossless with respect to information transform rate* $s(\hat{d}, \bar{r})$.

Proof Let

$$V(\bar{e}(k)) = \bar{e}^T(k)P_o(k)\bar{e}(k),$$

where $P_o(k) \geq 0$ is the solution of Riccati recursion (7.17), and consider

$$V(\bar{e}(k+1)) - V(\bar{e}(k)).$$

It follows from (7.41) that

$$\begin{aligned}
&V\left(\bar{e}(k+1)\right)-V\left(\bar{e}(k)\right)\\
&=\bar{e}^T(k+1)\left(P_o\left(k+1\right)-\bar{A}_o(k)P_o\left(k\right)\bar{A}_o^T\left(k\right)\right)\bar{e}(k+1)\\
&\quad-2\bar{e}^T(k+1)\bar{A}_o(k)P_o\left(k\right)\bar{C}_o^T(k)\bar{r}(k+1)-\bar{r}^T(k+1)\bar{C}_o(k)P_o\left(k\right)\bar{C}_o^T(k)\bar{r}(k+1).
\end{aligned}$$

By means of Riccati recursion (7.17) and relation (7.21), it holds

$$\begin{aligned}
&V\left(\bar{e}(k+1)\right)-V\left(\bar{e}(k)\right)\\
&=\bar{e}^T(k+1)\bar{E}_{d,o}(k)\bar{E}_{d,o}^T(k)\bar{e}(k+1)+2\bar{e}^T(k+1)\bar{E}_{d,o}(k)\bar{F}_{d,o}^T\bar{r}(k+1)\\
&\quad-\bar{r}^T(k+1)\bar{C}_o(k)P_o\left(k\right)\bar{C}_o^T(k)\bar{r}(k+1)\\
&=\hat{d}^T(k+1)\hat{d}(k+1)-\bar{r}^T(k+1)\bar{r}(k+1),\\
&\text{with } V_o(k)\left(C(k)P_o\left(k\right)C^T(k)+F_d(k)F_d^T(k)\right)V_o^T(k)=I.
\end{aligned}$$

Finally, we have

$$\begin{aligned}
&\sum_{k=0}^{N}\left(V\left(\bar{e}(k+1)\right)-V\left(\bar{e}(k)\right)\right)\\
&=-V\left(\bar{e}(0)\right)=\sum_{k=0}^{N}\left(\hat{d}^T(k+1)\hat{d}(k+1)-\bar{r}^T(k+1)\bar{r}(k+1)\right)\Longleftrightarrow\\
&\left\|\bar{r}(k+1)\right\|_{2.[0,N]}^2=V\left(\bar{e}(0)\right)+\left\|\hat{d}(k+1)\right\|_{2,[0,N]}^2.
\end{aligned}$$

The theorem is proved.

Theorem 7.3 gives a deeper insight into the unified solution and reveals its lossless property with respect to the information transform rate $s(\hat{d},\bar{r})$. This property can be, for instance, applied to developing model-free algorithms to achieve the unified solution. To this end, we can start, on the assumption of zero input vector, with a deadbeat system (of the appropriate dimension) as an observer-based residual generator and drive its dual system using the generated residual vector. It follows an update of the dual system, that is $L(k)$, $V(k)$, to minimise

$$\left\|\bar{r}(k+1)\right\|_{2,[0,N]}^2-V\left(\bar{e}(0)\right)-\left\|\hat{d}(k+1)\right\|_{2,[0,N]}^2.$$

Repeat these steps until

$$\left\|\bar{r}(k+1)\right\|_{2,[0,N]}^2\simeq V\left(\bar{e}(0)\right)+\left\|\hat{d}(k+1)\right\|_{2,[0,N]}^2.$$

7.5 Operator-Based Co-inner-outer Factorisation and Interpretations

In the previous sections, we have derived and investigated the unified solution on the basis of the algebraic input-output models. On the other hand, it is the state of the art in the fault detection research that the design of observer-based FD systems is approached by a multi-objective optimisation of the residual generator aiming at a trade-off between the system robustness and fault sensitivity. We now briefly study the unified solution in this context. For our purpose, we will apply the operator theory as the mathematical tool.

7.5.1 Operator-Based Models of the Residual Dynamics

We first introduce some definitions and mathematical preliminaries for the system representation using a linear operator.

In the fault-free case, the dynamics of the residual generator given by (7.5)–(7.6) defines a linear operator $\mathcal{H}_{d,[0,N]}$ that maps $(e(0), d)$ to r. That is

$$\mathcal{H}_{d,[0,N]} : \mathcal{R}^n \times l_{2,[0,N]} \to l_{2,[0,N]}, \tag{7.51}$$
$$r_d(k) = \bar{C}(k)\Phi(k,0)e(0) + \bar{C}(k)\sum_{j=0}^{k-1}\Phi(k,j+1)\bar{E}_d(j)d(j) + \bar{F}_d(k)d(k).$$

With the inner product in $\mathcal{R}^n \times l_{2,[0,N]}$ defined by

$$\langle (e_1, d_1), (e_2, d_2)\rangle_{2,[0,N]} = e_1^T e_2 + \sum_{k=0}^{N} d_1^T(k)\, d_2(k),$$

the adjoint of $\mathcal{H}_{d,[0,N]}$,

$$\mathcal{H}^*_{d,[0,N]} : l_{2,[0,N]} \to \mathcal{R}^n \times l_{2,[0,N]},$$

can be determined by

$$\left\langle r_d, \mathcal{H}_{d,[0,N]}(e,d)\right\rangle_{2,[0,N]} = \left\langle \mathcal{H}^*_{d,[0,N]} r_d, (e,d)\right\rangle_{2,[0,N]}.$$

It turns out

$$\mathcal{H}^*_{d,[0,N]} r_d = \begin{bmatrix} \sum_{i=0}^{N} \Phi^T(i,0)\bar{C}^T(i) r_d(i) \\ \sum_{i=j+1}^{N} \bar{E}_d^T(j)\Phi^T(i,j+1)\bar{C}^T(i) r_d(i) + \bar{F}_d^T(j) r_d(j) \end{bmatrix}. \tag{7.52}$$

Similarly, for the analysis of the influence of the faults on the residual signal, we define a linear operator $\mathcal{H}_{f,[0,N]}$ that maps f to r:

$$\begin{aligned} &\mathcal{H}_{f,[0,N]} : l_{2,[0,N]} \to l_{2,[0,N]}, \\ &\mathcal{H}_{f,[0,N]}\left(f\right) : r_f(k) = \bar{C}(k) \sum_{j=0}^{k-1} \Phi(k, j+1)\bar{E}_f(j) f(j) + \bar{F}_f(k) f(k), \end{aligned} \tag{7.53}$$

whose state-space representation is given by (7.5)–(7.6) for $d(k) = 0$. The adjoint of $\mathcal{H}_{f,[0,N]}$ is

$$\mathcal{H}^*_{f,[0,N]} r_f = \sum_{i=j+1}^{N} \bar{E}_f^T(j)\Phi^T(i, j+1)\bar{C}^T(i) r_f(i) + \bar{F}_f^T(j) r_f(j). \tag{7.54}$$

Using the notation introduced in (7.11)–(7.13), $\mathcal{H}_{d,[0,N]}$, $\mathcal{H}_{f,[0,N]}$ and their adjoints can be re-written into

$$\begin{aligned} \mathcal{H}_{d,[0,N]}\left(e, d\right)(k) &= g_e(k, 0)e(0) + \sum_{j=0}^{k} g_d(k, j)d(j), \\ \mathcal{H}^*_{d,[0,N]} r_d &= \begin{bmatrix} \sum_{i=0}^{N} g_e^T(i, 0) r_d(i) \\ \sum_{i=j}^{N} g_d^T(i, j) r_d(i) \end{bmatrix}, \\ \mathcal{H}_{f,[0,N]} f(k) &= \sum_{j=0}^{k} g_f(k, j) f(j), \\ \mathcal{H}^*_{f,[0,N]} r_f &= \sum_{i=j}^{N} g_f^T(i, j) r_f(i). \end{aligned}$$

The overall residual dynamics is modelled by

$$r(k) = \mathcal{H}_{d,[0,N]}\left(e, d\right)(k) + \mathcal{H}_{f,[0,N]} f(k). \tag{7.55}$$

Similar to our previous study, we can view the dynamics of the system output $y(k)$ with respect to the disturbance $d(k)$ and fault $f(k)$ as a special case of the residual dynamics (7.55) with

$$V(k) = I, L(k) = 0.$$

We denote them by

$$\begin{aligned} &y_d(k) = r_d(k) = \mathcal{H}_{y_d,[0,N]}\left(e, d\right)(k), \, y_f(k) = r_f(k) = \mathcal{H}_{y_f,[0,N]}\left(f\right)(k), \\ &\mathcal{H}_{y_d,[0,N]} = \mathcal{H}_{d,[0,N]}\left|_{L(k)=0, V(k)=I}\right., \mathcal{H}_{y_f,[0,N]} = \mathcal{H}_{f,[0,N]}\left|_{L(k)=0, V(k)=I}\right.. \end{aligned} \tag{7.56}$$

7.5.2 Co-inner-outer Factorisation and the Unified Solution

We now give an "operator version" of the unified solution using a co-inner-outer factorisation of the defined linear operator. A co-inner-outer factorisation of an operator is described as follows.

Definition 7.2 *An operator $\mathcal{H}_{ci,[0,N]} : l_{2,[0,N]} \to l_{2,[0,N]}$ is called co-inner, when $\forall \xi(k) \in l_{2,[0,N]}$*

$$\left\langle \xi, \mathcal{H}_{ci,[0,N]} \mathcal{H}^{*}_{ci,[0,N]} \xi \right\rangle_{2,[0,N]} = \sum_{k=0}^{N} \xi^{T}(k) \xi(k) = \langle \xi, \xi \rangle_{2,[0,N]} .$$

A factorisation of the operator $\mathcal{H}_{[0,N]} : l_{2,[0,N]} \to l_{2,[0,N]}$

$$\mathcal{H}_{[0,N]} = \mathcal{H}_{co,[0,N]} \circ \mathcal{H}_{ci,[0,N]}$$

is called co-inner-outer factorisation, when $\mathcal{H}_{ci,[0,N]}$ is co-inner and $\mathcal{H}_{co,[0,N]}$ is invertible which is called co-outer.

For our purpose, we first define a linear operator $\mathcal{Q}_{[0,N]} : l_{2,[0,N]} \to l_{2,[0,N]}$ with

$$\mathcal{Q}_{[0,N]} r(k) = C_{Q}(k) \sum_{j=0}^{k-1} \Phi_{Q}(k, j+1) B_{Q}(j) r(j) + D_{Q}(k) r(k) \tag{7.57}$$

for $k \in [0, N]$, where $r(k)$ is a residual vector generated by an observer-based residual generator, and $C_{Q}(k)$, $B_{Q}(j)$, (invertible) $D_{Q}(k)$ and $\Phi_{Q}(k, j+1)$ are some matrices of the appropriate dimensions. It is obvious that $\mathcal{Q}_{[0,N]}$ defines a dynamic post-filter driven by $r(k)$. Now, we are able to introduce the following lemma, which is similar to Lemma 7.2 and plays an important role in our subsequent study.

Lemma 7.4 *Given residual vectors*

$$r_{1}(k) = \mathcal{H}^{1}_{d,[0,N]} (e, d) (k) + \mathcal{H}^{1}_{f,[0,N]} (f) (k), \tag{7.58}$$

$$r_{2}(k) = \mathcal{H}^{2}_{d,[0,N]} (e, d) (k) + \mathcal{H}^{2}_{f,[0,N]} (f) (k), \tag{7.59}$$

where $r_{1}(k), r_{2}(k)$ are generated by two observer-based residual generators with (regular) $V_{1}(k)$, $V_{2}(k)$ and observer gain matrices $L_{1}(k)$, $L_{2}(k)$, respectively, then it holds

$$\begin{aligned}
r_1(k) &= \mathcal{Q}_{[0,N]} r_2(k) \qquad (7.60)\\
&= C_Q(k) \sum_{j=0}^{k-1} \Phi_Q(k, j+1) B_Q(j) r_2(j) + D_Q(k) r_2(k),\\
C_Q(k) &= V_1(k) C(k),\ D_Q(k) = V_1(k) V_2^{-1}(k),\\
B_Q(j) &= (L_1(j) - L_2(j))\, V_2^{-1}(j), 0 \leq j < k,\\
\Phi_Q(k, j+1) &= \prod_{l=j+1}^{k-1} (A(l) - L_1(l)C(l))\,, 0 \leq j < k-1, \Phi_Q(k,k) = I.
\end{aligned}$$

Proof According to (7.51) and (7.53), $r_i(k), i = 1, 2,$ given in (7.58)–(7.59) can be re-written into

$$\begin{aligned}
r_i(k) &= \mathcal{H}^i_{d,[0,N]}\,(e,d)\,(k) + \mathcal{H}^i_{f,[0,N]} f(k)\\
&= g^i_e(k,0)e(0) + \sum_{j=0}^{k} g^i_d(k,j)d(j) + \sum_{j=0}^{k} g^i_f(k,j)f(j), i = 1, 2,\\
g^i_e(k,0) &= g_e(k,0)\left|_{L(k)=L_i(k), V(k)=V_i(k)}\right. ,\\
g^i_d(k,j) &= g_d(k,j)\left|_{L(k)=L_i(k), V(k)=V_i(k)}\right. ,\\
g^i_f(k,j) &= g_f(k,j)\left|_{L(k)=L_i(k), V(k)=V_i(k)}\right. , i = 1, 2.
\end{aligned}$$

It follows from the lines in the proof of Lemma 7.2 and (7.25) that

$$\begin{aligned}
& g^1_e(k,0)e(0) + \sum_{j=0}^{k} g^1_d(k,j)d(j)\\
&\quad = C_Q(k) \sum_{j=0}^{k-1} \Phi_Q(k,j+1) B_Q(j) \left(g^2_e(j,0)e(0) + \sum_{i=0}^{j} g^2_d(j,i)d(i) \right)\\
&\quad + D_Q(k) \left(g^2_e(k,0)e(0) + \sum_{i=0}^{k} g^2_d(k,i)d(i) \right),\\
& \sum_{j=0}^{k} g^1_f(k,j)f(j) = C_Q(k) \sum_{j=0}^{k-1} \Phi_Q(k,j+1) B_Q(j) \sum_{i=0}^{j} g^2_f(j,i)f(i)\\
&\quad + D_Q(k) \sum_{i=0}^{k} g^2_f(k,i)f(i),
\end{aligned}$$

which results in

$$r_1(k) = C_Q(k) \sum_{j=0}^{k-1} \Phi_Q(k, j+1) B_Q(j) r_2(j) + D_Q(k) r_2(k).$$

Thus, (7.60) is proved.

Note that (7.60) can also be equivalently written as

$$\begin{aligned}
r_2(k) &= \mathcal{Q}_{[0,N]}^{-1} r_1(k) =: \bar{\mathcal{Q}}_{[0,N]} r_1(k), \\
\bar{\mathcal{Q}}_{[0,N]} r_1(k) &= C_{\bar{Q}}(k) \sum_{j=0}^{k-1} \Phi_{\bar{Q}}(k, j+1) B_{\bar{Q}}(j) r_1(j) + D_{\bar{Q}}(k) r_1(k), \\
C_{\bar{Q}}(k) &= V_2(k) C(k), D_{\bar{Q}}(k) = V_2(k) V_1^{-1}(k), \\
B_{\bar{Q}}(j) &= \left(L_2(j) - L_1(j)\right) V_1^{-1}(j), 0 \leq j < k, \\
\Phi_{\bar{Q}}(k, j+1) &= \prod_{l=j+1}^{k-1} \left(A(l) - L_2(l) C(l)\right), 0 \leq j < k-1, \Phi_{\bar{Q}}(k, k) = I.
\end{aligned}$$

Moreover, it holds

$$\begin{aligned}
\mathcal{H}_{d,[0,N]}^1 (e, d) &= \mathcal{Q}_{[0,N]} \circ \mathcal{H}_{d,[0,N]}^2 (e, d), \\
\mathcal{H}_{f,[0,N]}^1 (f) &= \mathcal{Q}_{[0,N]} \circ \mathcal{H}_{f,[0,N]}^2 (f).
\end{aligned}$$

The above results allow us to factorise $\mathcal{H}_{y_d,[0,N]}$ as

$$\begin{aligned}
\mathcal{H}_{y_d,[0,N]} &= \mathcal{Q}_{[0,N]} \circ \mathcal{H}_{d,o,[0,N]}, \\
\mathcal{H}_{d,o,[0,N]} &= \mathcal{H}_{d,[0,N]} \Big|_{L(k)=L_o(k), V(k)=V_o(k)},
\end{aligned}$$

where $L_o(k)$, $V_o(k)$ satisfy (7.15) and (7.16), respectively, and the post-filter $\mathcal{Q}_{[0,N]}$, as defined in (7.57), is invertible and given by

$$C_Q(k) = C(k), D_Q(k) = V_o^{-1}(k), B_Q(j) = -L_o(j) V_o^{-1}(j), 0 \leq j < k, \tag{7.61}$$

$$\Phi_Q(k, j+1) = \prod_{l=j+1}^{k-1} A(l), 0 \leq j < k-1, \Phi_Q(k, k) = I. \tag{7.62}$$

On the other hand, it follows from Lemma 7.1 that

$$\begin{aligned}&\left\langle r_d, \mathcal{H}_{d,o,[0,N]}\mathcal{H}^*_{d,o,[0,N]}r_d\right\rangle_{2,[0,N]}\\ &=\sum_{k=0}^{N} r_d^T(k)\left(g_{e,o}(k,0)g_{e,o}^T(k,0)+\sum_{j=0}^{k} g_{d,o}(k,j)g_{d,o}^T(k,j)\right)r_d(k)\\ &=\sum_{k=0}^{N} r_d^T(k)r_d(k)=\langle r_d, r_d\rangle_{2,[0,N]}\,.\end{aligned}$$

In order words, $\mathcal{H}_{d,o,[0,N]}$ is co-inner. In summary, we have proved the following theorem.

Theorem 7.4 *Given $\mathcal{H}_{y_d,[0,N]}$, as defined in (7.56), then*

$$\mathcal{H}_{y_d,[0,N]}=\mathcal{Q}_{[0,N]}\circ\mathcal{H}_{d,o,[0,N]} \tag{7.63}$$

is a co-inner-outer factorisation of $\mathcal{H}_{y_d,[0,N]}$ with $\mathcal{H}_{d,o,[0,N]}$ being co-inner and $\mathcal{Q}_{[0,N]}$ defined in (7.61)–(7.62) being co-outer.

As a result, the optimal residual generator design can be interpreted as finding a post-filter, which is the inverse of the co-outer. That is,

$$\begin{aligned}r(k)&=\mathcal{Q}^{-1}_{[0,N]}y_d(k)=\mathcal{H}_{d,o,[0,N]}\,(e,d)\,(k),\\ \mathcal{Q}^{-1}_{[0,N]}y_d(k)&=C_{\bar{Q}}(k)\sum_{j=0}^{k-1}\Phi_{\bar{Q}}(k,j+1)B_{\bar{Q}}(j)y_d(j)+D_{\bar{Q}}(k)y_d(k),\\ C_{\bar{Q}}(k)&=V_o(k)C(k),\,D_{\bar{Q}}(k)=V_o(k),\,B_{\bar{Q}}(j)=L_o(j),0\le j<k,\\ \Phi_{\bar{Q}}(k,j+1)&=\prod_{l=j+1}^{k-1}\left(A(l)-L_o(l)C(l)\right),0\le j<k-1,\Phi_{\bar{Q}}(k,k)=I.\end{aligned}$$

Next, we briefly discuss the unified solution from the system theoretic viewpoint briefly. Recall

$$\mathcal{H}_{d,o,[0,N]}\mathcal{H}^*_{d,o,[0,N]}=\mathcal{Q}^{-1}_{[0,N]}\mathcal{H}_{y,d,[0,N]}\left(\mathcal{Q}^{-1}_{[0,N]}\mathcal{H}_{y,d,[0,N]}\right)^*=I, \tag{7.64}$$

$$r(k)=\mathcal{Q}^{-1}_{[0,N]}\mathcal{H}_{y,d,[0,N]}\,(e,d)\,(k)+\mathcal{Q}^{-1}_{[0,N]}\mathcal{H}_{y,f,[0,N]}f(k). \tag{7.65}$$

Since $\mathcal{H}_{y,d,[0,N]}\mathcal{H}^*_{y,d,[0,N]}$ can be interpreted as the magnitude profile of the influence of $x(0), d(k)$ on $y(k)$, (7.64) means that the operator $\mathcal{Q}^{-1}_{[0,N]}$ is an "inverse" of the magnitude profile of $\mathcal{H}_{y,d,[0,N]}$. Moreover, we see from (7.65) that the operator $\mathcal{Q}^{-1}_{[0,N]}$ acts as a weighting on the influence of $f(k)$ on the residual signal. As a result, we claim that the unified solution can be interpreted as weighting the influence of $f(k)$ on $r(k)$ by using the "inverse" of the magnitude profile of the influence of $x(0), d(k)$ on $y(k)$.

7.5.3 Robustness Vs. Sensitivity

Due to their close relation to observers, observer-based residual generators are often designed in the context of the trade-off between robustness against disturbances and sensitivity to the faults. Blow, we briefly demonstrate that the unified solution given above also delivers an optimal trade-off in such a context. To this end, we first introduce necessary definitions and formulate our problems to be addressed.

Definition 7.3 *Let the linear operator $\mathcal{H}_{d,[0,N]} : \mathcal{R}^n \times l_{2,[0,N]}$, be given in (7.51). The l_2-gain of $\mathcal{H}_{d,[0,N]}$ is defined as*

$$\left\|\mathcal{H}_{d,[0,N]}\right\|_2 = \sup_{d\in l_{2,[0,N]}, e(0)\in\mathcal{R}^n} \frac{\|r_d(k)\|_{2,[0,N]}}{\sqrt{\|d(k)\|_{2,[0,N]}^2 + \|e(0)\|^2}}. \tag{7.66}$$

Considering that the fault could be present in any direction in the residual subspace, we use $\mathcal{H}^*_{f,[0,N]}\mathcal{H}_{f,[0,N]}$, instead of a norm or a scalar function of $\mathcal{H}_{f,[0,N]}$, to measure the fault sensitivity. Given two residual generators with $L_1(k)$, $V_1(k)$ and $L_2(k)$, $V_2(k)$, respectively, we say that the residual generator with $L_1(k)$, $V_1(k)$ offers a better sensitivity-to-robustness trade-off than the residual generator with $L_2(k)$, $V_2(k)$ if $\forall f(k)$

$$\frac{\left\langle\mathcal{H}_{f,1,[0,N]}f, \mathcal{H}_{f,1,[0,N]}f\right\rangle_{2,[0,N]}}{\left\|\mathcal{H}_{d,1,[0,N]}\right\|_2^2} - \frac{\left\langle\mathcal{H}_{f,2,[0,N]}f, \mathcal{H}_{f,2,[0,N]}f\right\rangle_{2,[0,N]}}{\left\|\mathcal{H}_{d,2,[0,N]}\right\|_2^2} \geq 0, \tag{7.67}$$
$$\mathcal{H}_{f,i,[0,N]} = \mathcal{H}_{f,[0,N]}\left|_{L(k)=L_i(k),V(k)=V_i(k)}\right.,$$
$$\mathcal{H}_{d,i,[0,N]} = \mathcal{H}_{d,[0,N]}\left|_{L(k)=L_i(k),V(k)=V_i(k)}\right., i = 1, 2.$$

For the sake of simplification, we introduce the following notation.

Definition 7.4 *The fact that $\forall f(k)$ (7.67) holds is denoted by*

$$\frac{\mathcal{H}^*_{f,1,[0,N]}\mathcal{H}_{f,1,[0,N]}}{\left\|\mathcal{H}_{d,1,[0,N]}\right\|_2^2} \geq \frac{\mathcal{H}^*_{f,2,[0,N]}\mathcal{H}_{f,2,[0,N]}}{\left\|\mathcal{H}_{d,2,[0,N]}\right\|_2^2} \text{ or} \tag{7.68}$$
$$\frac{\mathcal{H}^*_{f,1,[0,N]}\mathcal{H}_{f,1,[0,N]}}{\left\|\mathcal{H}_{d,1,[0,N]}\right\|_2^2} - \frac{\mathcal{H}^*_{f,2,[0,N]}\mathcal{H}_{f,2,[0,N]}}{\left\|\mathcal{H}_{d,2,[0,N]}\right\|_2^2} \geq 0. \tag{7.69}$$

We now formulate the optimal trade-off fault detection problem as an $\mathcal{H}_f/l_2$ optimisation problem.

Definition 7.5 *($\mathcal{H}_f/l_2$ optimisation) Given the residual generator (7.3)–(7.4), find, if exist, $L(k)$, $V(k)$ which maximise the (matrix-valued) ratio*

$$J_{f/2} = \frac{\mathcal{H}^*_{f,[0,N]}\mathcal{H}_{f,[0,N]}}{\left\|\mathcal{H}_{d,[0,N]}\right\|_2^2}. \tag{7.70}$$

Remark 7.3 *The $\mathcal{H}_f/l_2$ optimisation problem can be interpreted as a general form of the H_i/H_∞ optimisation problem formulated in (4.44) for LTI systems.*

Remark 7.4 *It should be pointed out for any two residual generators with $L_1(k)$, $V_1(k)$ and $L_2(k)$, $V_2(k)$, the matrix-valued ratios,*

$$\frac{\mathcal{H}^*_{f,1,[0,N]}\mathcal{H}_{f,1,[0,N]}}{\left\|\mathcal{H}_{d,1,[0,N]}\right\|_2^2} \text{ and } \frac{\mathcal{H}^*_{f,2,[0,N]}\mathcal{H}_{f,2,[0,N]}}{\left\|\mathcal{H}_{d,2,[0,N]}\right\|_2^2}$$

may be incomparable. Thus, the proof of the existence of the $\mathcal{H}_f/l_2$ optimisation is necessary. If it is solvable and let $\bar{L}(k)$, $\bar{V}(k)$ be the optimal solution of (7.70), then the $\mathcal{H}_f/l_2$ optimisation can be understood that for any given $L(k)$, $V(k)$ we have: $\forall f(k)$

$$\frac{\left\langle \bar{\mathcal{H}}_{f,[0,N]}f, \bar{\mathcal{H}}_{f,[0,N]}f\right\rangle}{\left\|\bar{\mathcal{H}}_{d,[0,N]}\right\|_2^2} \geq \frac{\left\langle \mathcal{H}_{f,[0,N]}f, \mathcal{H}_{f,[0,N]}f\right\rangle}{\left\|\mathcal{H}_{d,[0,N]}\right\|_2^2}, \tag{7.71}$$
$$\bar{\mathcal{H}}_{f,[0,N]} = \mathcal{H}_{f,[0,N]}\left|_{L(k)=\bar{L}(k),V(k)=\bar{V}(k)}\right.,$$
$$\bar{\mathcal{H}}_{d,[0,N]} = \mathcal{H}_{d,[0,N]}\left|_{L(k)=\bar{L}(k),V(k)=\bar{V}(k)}\right..$$

In this context, $\bar{L}(k)$, $\bar{V}(k)$ are understood as the best trade-off solution. We would like to emphasise that the inequality (7.71) should hold for all possible faults. In fact, for this reason, we have introduced the matrix-valued ratio (7.71).

Before presenting the solution to the $\mathcal{H}_f/l_2$ optimisation problem, we introduce some necessary preliminary results. Generally, given a linear operator $\mathcal{H}_{[0,N]}$ that maps $\varsigma(k) \in l_{2,[0,N]}$ to $\xi(k) \in l_{2,[0,N]}$ and whose l_2-gain is defined by

$$\begin{aligned}\left\|\mathcal{H}_{[0,N]}\right\|_2 &= \sup_{\varsigma\in l_{2,[0,N]},\varsigma\neq 0}\frac{\|\xi(k)\|_{2,[0,N]}}{\|\varsigma(k)\|_{2,[0,N]}}\\ &= \sup_{\varsigma\in l_{2,[0,N]},\varsigma\neq 0}\frac{\left\|\mathcal{H}_{[0,N]}\varsigma(k)\right\|_{2,[0,N]}}{\|\varsigma(k)\|_{2,[0,N]}},\end{aligned}$$

it holds

$$\left\|\mathcal{H}_{[0,N]}\right\|_2 = \left\|\mathcal{H}^*_{[0,N]}\right\|_2.$$

Note that

$$\left\|\mathcal{H}^*_{[0,N]}\right\|_2^2 = \sup_{\xi\in l_{2,[0,N]},\xi\neq 0}\frac{\left\langle \mathcal{H}^*_{[0,N]}\xi(k), \mathcal{H}^*_{[0,N]}\xi(k)\right\rangle_{2,[0,N]}}{\|\xi(k)\|_2^2}, \tag{7.72}$$
$$\left\langle \mathcal{H}^*_{[0,N]}\xi(k), \mathcal{H}^*_{[0,N]}\xi(k)\right\rangle_{2,[0,N]} = \left\langle \xi(k), \mathcal{H}_{[0,N]}\mathcal{H}^*_{[0,N]}\xi(k)\right\rangle_{2,[0,N]}. \tag{7.73}$$

Equations (7.72)–(7.73) will be used for the l_2-gain computation of an operator.

Remark 7.5 *In the above discussion, no restriction has been made on the time interval* $[0, N]$. *Notice that for a finite* N, *the* l_2*-gain of* $\mathcal{H}_{d,[0,N]}$ *and* $\mathcal{H}^*_{f,[0,N]}\mathcal{H}_{f,[0,N]}$ *can be equivalently expressed in terms of* $\bar{\sigma}\left(H_{\bar{d}}(0, N)\right)$ *and* $H^T_{\bar{f}}(0, N)H_{\bar{f}}(0, N)$, *respectively. With the introduction of the above operators, we are also able to address the case with* $[0, N] = [0, \infty]$. *It should be pointed out that a fault may be unbounded for* $N \longrightarrow \infty$. *On the other hand, it is of practical interest that a fault is detected in a short finite time after its occurrence. Under this consideration, it is supposed that*

$$f(k) = \begin{cases} f(k), k \leq K, \\ 0, k > K, \end{cases} \quad K(< \infty) \text{ is a (large) integer,}$$

so that $f(k)$ *is also* $l_{2,[0,\infty]}$*-bounded without loss of the practical applicability.*

We are now in a position to present the results on the $\mathcal{H}_f/l_2$ optimisation problem.

Theorem 7.5 *Given the system model (7.1)–(7.2) and residual generator (7.5)–(7.4), the* $\mathcal{H}_f/l_2$ *optimisation problem can be solved by*

$$L(k) = L_o(k),\ V(k) = \beta V_o(k)$$

with any (real) constant β *and* $L_o(k)$, $V_o(k)$ *given in (7.15)–(7.16) in Lemma 7.1.*

Proof Recall

$$\begin{aligned} &\left\langle r_d, \mathcal{H}_{d,[0,N]}\mathcal{H}^*_{d,[0,N]}r_d\right\rangle_{2,[0,N]} \\ &= \sum_{k=0}^{N} r_d^T(k) \left(g_e(k,0) \sum_{i=0}^{k} g_e^T(i,0) r_d(i) + \sum_{j=0}^{k} g_d(k,j) \sum_{i=j}^{k} g_d^T(i,j) r_d(i) \right), \end{aligned}$$

and for $L(k) = L_o(k)$, $V(k) = V_o(k)$, it holds

$$\left\langle r_d, \mathcal{H}_{d,o,[0,N]}\mathcal{H}^*_{d,o,[0,N]}r_d\right\rangle_{2,[0,N]} = \sum_{k=0}^{N} r_d^T(k) r_d(k). \tag{7.74}$$

Noting that (7.74) means for $L(k) = L_o(k)$, $V(k) = V_o(k)$,

$$\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2 = \left\| \mathcal{H}^*_{d,o,[0,N]} \right\|_2 = 1,$$

we have

$$J_{f/2} = \frac{\mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o,[0,N]}}{\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2^2} = \mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o,[0,N]}. \tag{7.75}$$

For any $L(k)$, $V(k)$ different from $L_o(k)$, $V_o(k)$, it holds, according to Lemma 7.4

$$\begin{aligned}
&\left\langle r_d, \mathcal{H}_{d,[0,N]}\mathcal{H}^*_{d,[0,N]}r_d\right\rangle_{2,[0,N]} = \left\langle r_d, \mathcal{Q}_{[0,N]}\mathcal{Q}^*_{[0,N]}r_d\right\rangle_{2,[0,N]} \Longleftrightarrow \\
&\left\|\mathcal{H}_{d,[0,N]}\right\|_2^2 = \left\|\mathcal{H}^*_{d,[0,N]}\right\|_2^2 \\
&= \sup_{r_d \in l_{2,[0,N]}} \frac{\left\langle r_d, \mathcal{H}_{d,[0,N]}\mathcal{H}^*_{d,[0,N]}r_d\right\rangle_{2,[0,N]}}{\|r_d\|_2^2} \\
&= \sup_{r_d \in l_{2,[0,N]}} \frac{\left\langle r_d, \mathcal{Q}_{[0,N]}\mathcal{Q}^*_{[0,N]}r_d\right\rangle_{2,[0,N]}}{\|r_d\|_2^2} = \left\|\mathcal{Q}^*_{[0,N]}\right\|_2^2 = \left\|\mathcal{Q}_{[0,N]}\right\|_2^2,
\end{aligned}$$

and moreover $\forall f(k)$

$$\begin{aligned}
&r_f(k) = \mathcal{Q}_{[0,N]}r_{f,o}(k) \Longrightarrow \left\langle \mathcal{H}_{f,[0,N]}f, \mathcal{H}_{f,[0,N]}f\right\rangle_{2,[0,N]} = \left\|r_f(k)\right\|_2^2 \\
&\leq \left\|\mathcal{Q}_{[0,N]}\right\|_\infty^2 \left\|r_{f,o}(k)\right\|_2^2 = \left\|\mathcal{Q}_{[0,N]}\right\|_\infty^2 \left\langle \mathcal{H}_{f,o,[0,N]}f, \mathcal{H}_{f,o,[0,N]}f\right\rangle_{2,[0,N]}.
\end{aligned}$$

That means, according to (7.75),

$$\begin{aligned}
\frac{\mathcal{H}^*_{f,[0,N]}\mathcal{H}_{f,[0,N]}}{\left\|\mathcal{Q}_{[0,N]}\right\|_\infty^2} &= \frac{\mathcal{H}^*_{f,[0,N]}\mathcal{H}_{f,[0,N]}}{\left\|\mathcal{H}_{d,[0,N]}\right\|_2^2} \leq \\
\mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o[0,N]} &= \frac{\mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o,[0,N]}}{\left\|\mathcal{H}_{d,o,[0,N]}\right\|_2^2}.
\end{aligned} \tag{7.76}$$

Thus, for any $L(k)$, $V(k)$, $\frac{\mathcal{H}^*_{f,[0,N]}\mathcal{H}_{f,[0,N]}}{\left\|\mathcal{H}_{d,[0,N]}\right\|_\infty^2}$ *is comparable with* $\frac{\mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o,[0,N]}}{\left\|\mathcal{H}_{d,o,[0,N]}\right\|_\infty^2}$ and, by Definition 7.4, smaller or equal to $\frac{\mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o,[0,N]}}{\left\|\mathcal{H}_{d,o,[0,N]}\right\|_\infty^2}$, which means that $L_o(k)$, $V_o(k)$ solve the $\mathcal{H}_f/l_2$ optimisation problem. Note that for any (real) constant β, the above results also hold for $V(k) = \beta V_o(k)$, since

$$\frac{\left(\beta\mathcal{H}_{f,o,[0,N]}\right)^* \beta\mathcal{H}_{f,o,[0,N]}}{\left\|\beta\mathcal{H}_{d,o,[0,N]}\right\|_\infty^2} = \frac{\mathcal{H}^*_{f,o,[0,N]}\mathcal{H}_{f,o,[0,N]}}{\left\|\mathcal{H}_{d,o,[0,N]}\right\|_\infty^2}.$$

As a result, the theorem is finally proved.

As introduced in Chap. 4, the so-called $\mathcal{H}_-/\mathcal{H}_\infty$ and $\mathcal{H}_\infty/\mathcal{H}_\infty$ optimisations are the well-established schemes for the residual generator design of LTI systems. Below, we demonstrate that they are the special cases of the $\mathcal{H}_f/l_2$ optimisation.

Definition 7.6 *$\mathcal{H}_-/l_2$ and l_2/l_2 optimisations are defined as: given the residual generator (7.3)–(7.4), find $L(k)$ and $V(k)$ so that the following cost functions,*

$$J_{-/2} = \frac{\left\| \mathcal{H}_{f,[0,N]} \right\|_{-}^{2}}{\left\| \mathcal{H}_{d,[0,N]} \right\|_{2}^{2}}, \left\| \mathcal{H}_{f,[0,N]} \right\|_{-} = \inf_{f \in l_{2,[0,N]}, f \neq 0} \frac{\left\| r_f(k) \right\|_2}{\| f(k) \|_2}, \tag{7.77}$$

$$J_{2/2} = \frac{\left\| \mathcal{H}_{f,[0,N]} \right\|_{2}^{2}}{\left\| \mathcal{H}_{d,[0,N]} \right\|_{2}^{2}}, \left\| \mathcal{H}_{f,[0,N]} \right\|_{\infty} = \sup_{f \in l_{2,[0,N]}, f \neq 0} \frac{\left\| r_f(k) \right\|_2}{\| f(k) \|_2}, \tag{7.78}$$

are respectively maximised.

Theorem 7.6 *Given the system model (7.1)–(7.2) and residual generator (7.5)–(7.4), then $L_o(k)$, $\beta V_o(k)$ with any (real) constant β and $L_o(k)$, $V_o(k)$ given in (7.15)–(7.16) unifiedly solve the $\mathcal{H}_- / l_2$ and l_2 / l_2 optimisation problems.*

Proof Re-write $J_{-/2}$, $J_{2/2}$ into

$$J_{-/2} = \frac{\inf_{\| f(k) \|_2 = 1} \left\| r_f(k) \right\|_2^2}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2}, J_{2/2} = \frac{\sup_{\| f(k) \|_2 = 1} \left\| r_f(k) \right\|_2}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2}$$

Let $f^1(k)$, $f^2(k)$ with $\left\| f^1(k) \right\|_2 = \left\| f^2(k) \right\|_2 = 1$ be the fault vectors satisfying

$$\left\langle \mathcal{H}_{f,o,[0,N]} f^1, \mathcal{H}_{f,o,[0,N]} f^1 \right\rangle = \left\| \mathcal{H}_{f,o,[0,N]} \right\|_{-}^{2},$$
$$\left\langle \mathcal{H}_{f,[0,N]} f^2, \mathcal{H}_{f,[0,N]} f^2 \right\rangle = \left\| \mathcal{H}_{f,[0,N]} \right\|_{2}^{2}.$$

Remember that $\forall f(k)$

$$\frac{\left\langle \mathcal{H}_{f,o,[0,N]} f, \mathcal{H}_{f,o,[0,N]} f \right\rangle}{\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2^2} = \left\| r_{f,o}(k) \right\|_2$$
$$\geq \frac{\left\langle \mathcal{H}_{f,[0,N]} f, \mathcal{H}_{f,[0,N]} f \right\rangle}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2} = \frac{\left\| r_f(k) \right\|_2}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2}.$$

As a result, we have

$$\frac{\left\| \mathcal{H}_{f,o,[0,N]} \right\|_{-}^{2}}{\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2^2} = \frac{\left\langle \mathcal{H}_{f,o,[0,N]} f^1, \mathcal{H}_{f,o,[0,N]} f^1 \right\rangle}{\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2^2}$$
$$\geq \frac{\left\langle \mathcal{H}_{f,[0,N]} f^1, \mathcal{H}_{f,[0,N]} f^1 \right\rangle}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2} = \frac{\left\| \mathcal{H}_{f,[0,N]} \right\|_{-}^{2}}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2},$$
$$\frac{\left\| \mathcal{H}_{f,o,[0,N]} \right\|_{2}^{2}}{\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2^2} = \frac{\left\langle \mathcal{H}_{f,o,[0,N]} f^2, \mathcal{H}_{f,o,[0,N]} f^2 \right\rangle}{\left\| \mathcal{H}_{d,o,[0,N]} \right\|_2^2}$$
$$\geq \frac{\left\langle \mathcal{H}_{f,[0,N]} f^2, \mathcal{H}_{f,[0,N]} f^2 \right\rangle}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2} = \frac{\left\| \mathcal{H}_{f,[0,N]} \right\|_{2}^{2}}{\left\| \mathcal{H}_{d,[0,N]} \right\|_2^2}.$$

Since for any constant β,

$$\frac{\left\|\beta \mathcal{H}_{f,o,[0,N]}\right\|_{-}^{2}}{\left\|\beta \mathcal{H}_{d,o,[0,N]}\right\|_{2}^{2}} = \frac{\left\|\mathcal{H}_{f,o,[0,N]}\right\|_{-}^{2}}{\left\|\mathcal{H}_{d,o,[0,N]}\right\|_{2}^{2}}, \frac{\left\|\beta \mathcal{H}_{f,o,[0,N]}\right\|_{2}^{2}}{\left\|\beta \mathcal{H}_{d,o,[0,N]}\right\|_{2}^{2}} = \frac{\left\|\mathcal{H}_{f,o,[0,N]}\right\|_{2}^{2}}{\left\|\mathcal{H}_{d,o,[0,N]}\right\|_{2}^{2}},$$

the theorem is thus proved.

Alternative to the $\mathcal{H}_{-}/l_2$ and l_2/l_2 optimisations, design problem formulated as

$$\max_{L,V} \left\|\mathcal{H}_{f,[0,N]}\right\| \text{ subject to } \left\|\mathcal{H}_{d,[0,N]}\right\|_{2} \leq \gamma \tag{7.79}$$

is often considered in the observer-based FD study on LTI systems, where $\gamma > 0$ is a given constant and $\left\|\mathcal{H}_{f,[0,N]}\right\|$ stands either for l_2-gain or $\mathcal{H}_{-}$ -index. This formulation allows, for example, the application of the well-established LMI multi-objective optimisation technique or the dynamic game theory to solving FD problems. The following theorem reveals that the unified solution also solves the above optimisation design problem.

Theorem 7.7 *Given the system model (7.1)–(7.2) and residual generator (7.5)–(7.4), then $L_o(k), \gamma V_o(k)$ with $L_o(k), V_o(k)$ given in (7.15)–(7.16) unifiedly solve (7.79), both for*

$$\left\|\mathcal{H}_{f,[0,N]}\right\| = \left\|\mathcal{H}_{f,[0,N]}\right\|_{2} \text{ or } \left\|\mathcal{H}_{f,[0,N]}\right\| = \left\|\mathcal{H}_{f,[0,N]}\right\|_{-}.$$

Proof Assume that for some $L(k) = \bar{L}(k), V(k) = \tilde{V}(k)$

$$\max_{L,V} \left\|\mathcal{H}_{f,[0,N]}\right\| = \left\|\tilde{\mathcal{H}}_{f,[0,N]}\right\|, \tilde{\mathcal{H}}_{f,[0,N]} = \mathcal{H}_{f,[0,N]} \left.\right|_{L(k)=\bar{L}(k), V(k)=\tilde{V}(k)}$$
$$\text{subject to } \left\|\tilde{\mathcal{H}}_{d,[0,N]}\right\|_{2} \leq \gamma, \tilde{\mathcal{H}}_{d,[0,N]} = \mathcal{H}_{d,[0,N]} \left.\right|_{L(k)=\bar{L}(k), V(k)=\tilde{V}(k)}.$$

Recall that

$$\left\|\gamma \mathcal{H}_{d,o,[0,N]}\right\|_{2} = \gamma$$

and $L_o(k), \gamma V_o(k)$ solve the H_{-}/l_2 and l_2/l_2 optimisations, which means

$$\frac{\left\|\tilde{\mathcal{H}}_{f,[0,N]}\right\|}{\left\|\tilde{\mathcal{H}}_{d,[0,N]}\right\|_{2}} \leq \frac{\left\|\gamma \mathcal{H}_{f,o,[0,N]}\right\|}{\left\|\gamma \mathcal{H}_{d,o,[0,N]}\right\|_{2}}.$$

Hence, it holds

$$\left\|\tilde{\mathcal{H}}_{f,[0,N]}\right\| = \max_{L,V} \left\|\mathcal{H}_{f,[0,N]}\right\| \leq \left\|\gamma \mathcal{H}_{f,o,[0,N]}\right\| \frac{\left\|\tilde{\mathcal{H}}_{d,[0,N]}\right\|_{\infty}}{\gamma} \leq \left\|\gamma \mathcal{H}_{f,o,[0,N]}\right\|.$$

Thus, $L_o(k), \gamma V_o(k)$ solve (7.79) and the theorem is proved.

Application of multi-objective optimisation techniques to solving (7.79) generally leads to an iterative computation of two Riccati inequalities. Differently, the unified solution only needs to solve a single Riccati recursion. Moreover, the unified solution solves $\mathcal{H}_-/l_2$ and l_2/l_2 optimisations simultaneously.

7.6 Examples

As examples, we apply the FD schemes proposed in this chapter to two special LDTV systems, the so-called switched systems and linear parameter varying (LPV) systems.

Example 7.1 *Consider the LDTV system model (7.1)–(7.2) with*

$$\begin{aligned} A(k) &= A_{\sigma(k)}, B(k) = B_{\sigma(k)}, C(k) = C_{\sigma(k)}, D(k) = D_{\sigma(k)}, \\ E_d(k) &= E_{d,\sigma(k)}, E_f(k) = E_{f,\sigma(k)}, F_d(k) = F_{d,\sigma(k)}, F_f(k) = F_{f,\sigma(k)}, \end{aligned}$$

where $\sigma(k) \in \{1, \cdots, M\}$ is called switching signal. It is a time function and indicates, for $\sigma(k) = i$,

$$\begin{aligned} A(k) &= A_i, B(k) = B_i, C(k) = C_i, D(k) = D_i, \\ E_d(k) &= E_{d,i}, E_f(k) = E_{f,i}, F_d(k) = F_{d,i}, F_f(k) = F_{f,i}. \end{aligned}$$

The LTI system

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + E_{d,i} d(k) + E_{f,i} f(k), \\ y(k) &= C_i x(k) + D_i u(k) + F_{d,i} d(k) + F_{f,i} f(k), \\ i &= 1, \cdots, M, \end{aligned}$$

is called the i-th sub-system. The switching signal $\sigma(k)$ and the system matrices of all M sub-systems are known. The above system model is called switched system.

In order to achieve an optimal fault detection, we propose to apply the following FDF

$$\begin{aligned} \hat{x}(k+1) &= A_{\sigma(k)}\hat{x}(k) + B_{\sigma(k)}u(k) + L(k)\left(y(k) - \hat{y}(k)\right), \\ r(k) &= V(k)\left(y(k) - \hat{y}(k)\right), \hat{y}(k) = C_{\sigma(k)}\hat{x}(k) + D_{\sigma(k)}u(k), \\ L(k) &= \left(A_{\sigma(k)}P(k)C^T_{\sigma(k)} + E_{d,\sigma(k)}F^T_{d,\sigma(k)}\right)V^2(k), \\ V(k) &= \left(C_{\sigma(k)}P(k)C^T_{\sigma(k)} + F_{d,\sigma(k)}F^T_{d,\sigma(k)}\right)^{-1/2}, \\ P(k+1) &= A_{\sigma(k)}P(k)A^T_{\sigma(k)} + E_{d,\sigma(k)}E^T_{d,\sigma(k)} - L(k)V^{-2}(k)L^T(k), \ P(0) = I, \end{aligned}$$

and the residual evaluation function

$$J_{2,[0,N]} = \sum_{k=0}^{N} r^T(k)r(k) = \|r(k)\|_{2,[0,N]}^2 .$$

Adopting the notations defined in Sect. 7.2, $J_{2,[0,N]}$ *can be expressed, during fault-free operations, by*

$$J_{2,[0,N]} = \bar{r}^T(0,N)\bar{r}(0,N) = \bar{d}^T(0,N)\, H_{\bar{d}}^T(0,N)H_{\bar{d}}(0,N)\bar{d}(0,N),$$
$$\bar{r}(0,N) = H_{\bar{d}}(0,N)\bar{d}(0,N).$$

It follows from Lemma 7.1 that $\forall N$

$$H_{\bar{d}}(0,N)H_{\bar{d}}^T(0,N) = I.$$

Thus, on the assumption

$$e^T(0)e(0) + \|r(k)\|_{2,[0,N]}^2 = \bar{d}^T(0,N)\,\bar{d}(0,N) \le \delta_{d,[0,k_2]}^2 + \delta_e^2 = \delta^2,$$

the threshold is set to be

$$J_{th} = \delta^2.$$

As proved in Theorem 7.1, the above FD system results in an optimal fault detection.

Example 7.2 *Substituting the system matrices in the LDTV system model (7.1)–(7.2) by*

$$A(k) = A(p(k)),\, B(k) = B(p(k)),\, C(k) = C(p(k)),\, D(k) = D(p(k)),$$
$$E_d(k) = E_d(p(k)),\, E_f(k) = E_f(p(k)),\, F_d(k) = F_d(p(k)),\, F_f(k) = F_f(p(k)),$$

leads to an LPV system model, where $p(k)$ *is the known, time-varying parameter vector. Based on it, we propose to apply the following FDF,*

$$\begin{aligned}
\hat{x}(k+1) &= A(p(k))\,\hat{x}(k) + B(p(k))u(k) + L(p(k))\left(y(k) - \hat{y}(k)\right),\\
r(k) &= V(p(k))\left(y(k) - \hat{y}(k)\right),\, \hat{y}(k) = C(p(k))\hat{x}(k) + D(p(k))u(k),\\
L(p(k))) &= \left(A(p(k))P(k)C^T(p(k)) + E_d(p(k))F_d^T(p(k))\right)V^2(p(k)),\\
V(p(k)) &= \left(C(p(k))P(k)C^T(p(k)) + F_d(p(k))F_d^T(p(k))\right)^{-1/2},\\
P(k+1) &= A(p(k))P(k)A^T(p(k)) + E_d(p(k))E_d^T(p(k))\\
&\quad - L(p(k))V^{-2}(p(k))L^T(p(k)),\, P(0) = I,
\end{aligned}$$

and the residual evaluation function

$$J_{2,[0,N]} = \sum_{k=0}^{N} r^T(k)r(k) = \|r(k)\|_{2,[0,N]}^2 .$$

With the same notation and argument given in Example 7.1, it holds, during fault-free operations,

$$J_{2,[0,N]} = \bar{r}^T(0,N)\bar{r}(0,N) = \bar{d}^T(0,N)\, H_{\bar{d}}^T(0,N)H_{\bar{d}}(0,N)\bar{d}(0,N),$$
$$\bar{r}(0,N) = H_{\bar{d}}(0,N)\bar{d}(0,N),\, H_{\bar{d}}(0,N)H_{\bar{d}}^T(0,N) = I.$$

As a result, setting the threshold equal to

$$J_{th} = \delta^2.$$

results in an optimal fault detection.

7.7 Notes and References

In this chapter, we have studied LDTV observer-based fault detection systems. This work is motivated not only by the recent development in the application domain but also by the research efforts since the last decade. In their work, Zhang and Ding [1], Li and Zhou [2] have systematically studied $\mathcal{H}_\infty/\mathcal{H}_\infty$ and $\mathcal{H}_-/\mathcal{H}_\infty$ fault detection problems for linear continuous time-varying systems (LTV) and provided an analytical solution. Zhang et al. [3, 4] and Li [5] have reported successful results on the FD in linear discrete time periodic and LDTV systems, respectively. In this decade, Prof. Zhong and her co-authors have been strongly active in the field of model-based fault detection for LDTV systems and made significant contributions to the theoretical framework. They have proposed numerous fault detection schemes using different techniques and methods, including time-varying Parity space methods [6–8], projection technique [9, 10] and optimisation technique [11] . Some of these and other existing methods have been well summarised in their recent survey paper [12].

We would like to mention that fault detection issues in LTV systems are often addressed, different from our study presented here, in the robustness framework, where the time varying parts in the process model are modelled either as (time varying) model uncertainties or as linear parameter varying (LPV) functions. For instance, in [13–15], the authors have proposed LMI-based approaches to deal with LTI systems with time-varying uncertainties, while in [16, 17] geometrical technique and LMI-based FD methods have been developed for the LPV systems. It is worth remarking that all these studies have been dedicated to the observer-based residual generator design in the $\mathcal{H}_\infty$ or $\mathcal{H}_-/\mathcal{H}_\infty$ trade-off framework.

Our work in this chapter consists of four parts. The first part has been dedicated to the formulation of the fault detection problems for LDTV systems in the context of the optimal fault detection problem formulated in Chap. 2. That means, the objective of the fault detection system design is to achieve maximal fault detectability. It is remarkable that four types of residual evaluation functions have been considered in our work.

After some mathematical efforts, the optimal solutions have been achieved and summarised in Theorems 7.1 and 7.2 in the second part of our work. It is of considerable interest to notice that

- for the four residual evaluation functions, the optimal fault detection systems (solutions) are comprised of the identical (optimal) residual generator. It is the optimal FDF, and
- the optimal FDF can be viewed as an extension of the unified solution presented in Chap. 4 to the LDTV systems. This aspect has been further studied in the third and fourth parts of our work.

Recall that co-inner-outer factorisation is an important interpretation and a tool as well for the unified solution. The achieved unified solution for LDTV systems has been intensively studied in this context. We have provided two different forms of the co-inner-outer factorisation for LDTV systems. Moreover, a deeper insight into the unified solution using the concept of information lossless is given, which is helpful for extending the achieved results to the data-driven and model-free framework.

In the last part of our work, we have studied the addressed fault detection problem in the classical “robustness vs. sensitivity” framework, in which the optimal fault detection is formulated as an optimisation problem with the ratio of the l_2-norm of the residual response to the unknown input to the l_2-norm of the residual response to the fault. To this end, the operator-based system model, the associated computations and handlings are involved. For the needed essential knowledge of operator theory, we refer the reader to [18]. It is demonstrated that the unified solution achieved in our work simultaneously solves the $\mathcal{H}_-/l_2$ and l_2/l_2 optimisation problems, which are the LDTV version of the well-known $\mathcal{H}_-/\mathcal{H}_\infty$ and $\mathcal{H}_\infty/\mathcal{H}_\infty$ optimisation problems for LTI-FDF design [19].

To conclude this chapter, we would like to remark that the LDTV state space representation is a very general model form of linear dynamic systems, to which system types like switched systems or LPV systems belong as well. We notice that few results have been reported on the application of LDTV system techniques to designing observer-based FD systems for switched or LPV systems, although it may result in optimal fault detection, as demonstrated in our examples given in the last section. In this regard, we would like to mention the work by Zhong et al. [20], in which the LDTV technique has been successfully applied to the solution of nonlinear fault detection problems.

References

1. P. Zhang and S. X. Ding, "Observer-based fault detection of linear time-varying systems (in German)," *Automatisierungstechnik*, vol. 52, pp. 370–376, 2004.
2. X. Li and K. Zhou, "A time domain approach to robust fault detection of linear time-varying systems," *Automatica*, vol. 45, pp. 94–102, 2009.
3. P. Zhang, S. Ding, G. Wang, and D. Zhou, "Fault detection of linear discrete-time periodic systems," *IEEE Trans. on Automatic Control*, vol. 50(2), pp. 239–244, 2005.
4. P. Zhang and S. X. Ding, "On fault detection in linear discrete-time, periodic, and sampled-data systems (survey)," *Journal of Control Science and Engineering*, pp. 1–18, 2008.
5. X. Li, *Fault Detection Filter Design for Linear Systems*.Thesis of Louisiana State University, 2009.
6. M. Zhong, S. X. Ding, Q. Han, and Q. Ding, "Parity space-based fault estimation for linear discrete time-varying systems," *IEEE Trans. on Autom. Contr.*, vol. 55, pp. 1726–1731, 2010.
7. M. Zhong, Y. Song, and S. X. Ding, "Parity space-based fault detection for linear discrete time-varying systems with unknown input," *Automatica*, vol. 59, pp. 120–126, 2015.
8. T. Xue, M. Zhong, S. X. Ding, and H. Ye, "Stationary wavelet transform aided design of parity space vectors for fault detection in LDTV systems," *IET Control Theory and Applications*, vol. 12, pp. 857–864, 2018.
9. M. Zhong, D. Zhou, and S. X. Ding, "On designing H_{inf} fault detection filter for linear discrete time-varying systems," *IEEE Trans. on Autom. Control*, vol. 55, pp. 1689–1695, 2010.
10. M. Zhong, S. X. Ding, and D. Zhou, "A new scheme of fault detection for linear discrete time-varying systems," *IEEE Trans. on Automat. Contr.*, vol. 61, pp. 2597–2602, 2016.
11. M. Zhong, S. X. Ding, and E. L. Ding, "Optimal fault detection for linear discrete time-varying systems," *Automatica*, vol. 46, pp. 1395–1400, 2010.
12. M. Zhong, T. Xue, and S. X. Ding, "A survey on model-based fault diagnosis for linear discrete time varying systems," *Neurocomputing*, vol. 306, pp. 51–60, 2018.
13. A. Edelmayer and J. Bokor, "Optimal h-infinity scaling for sensitivity optimization of detection filters," *Int. J. of Robust and Nonlinear Contr.*, vol. 12, pp. 749 – 760, 2002.
14. A. Casavola, D. Famularo, and G. Fraze, "A robust deconvolution scheme for fault detection and isolation of uncertain linear systems: An LMI approach," *Automatica*, vol. 41, pp. 1463–1472, 2005.
15. D. Henry and A. Zolghadri, "Design and analysis of robust residual generators for systems under feedback control," *Automatica*, vol. 41, pp. 251–264, 2005.
16. J. Bokor and G. Balas, "Detection filter design for LPV systems - a geometric approach," *Automatica*, vol. 40, pp. 511–518, 2004.
17. H. Wang and G.-H. Yang, "Integrated fault detection and control for LPV systems," *Int. J. of Robust and Nonlinear Control*, vol. 19, pp. 341–363, 2009.
18. F. M. Callier and C. A. Desoer, *Linear System Theory*. New York: Springer-Verlag, 1991.
19. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.
20. M. Zhong, L. Zhang, S. X. Ding, and D. Zhou, "A probabilistic approach to robust fault detection for a class of nonlinear systems," *IEEE Trans. on Indus. Elec.*, vol. 64, pp. 3930–3939, 2017.

Chapter 8
Fault Estimation in Linear Dynamic Systems

Fault estimation is a main issue in the fault diagnosis framework. As illustrated in previous chapters, fault estimation can be embedded in an optimal fault detection solution, and further delivers detailed information about the fault, after this fault is detected. In this chapter, we study the fault estimation problem defined in the following context.

Consider the LDTV system model

$$x(k+1) = A(k)x(k) + B(k)u(k) + E(k)f(k), \tag{8.1}$$
$$y(k) = C(k)x(k) + D(k)u(k) + F(k)f(k), \tag{8.2}$$

where $x(k) \in \mathcal{R}^n$, $u(k) \in \mathcal{R}^p$, $y(k) \in \mathcal{R}^m$ are system state, input and output vectors, respectively, $f(k) \in \mathcal{R}^{k_f}$ is the unknown fault vector to be estimated, and all system matrices are of appropriate dimensions and known. We assume that $f(k)$ can be any l_2-bounded time function. Thus, such a fault estimation problem can be, in general, formulated as unknown input estimation.

Before we begin with our study, we would like to call reader's attention to the following two facts in the fault detection scheme presented in the last chapter, which tell us that the unified solution and an optimal FDF can also be used for the purpose of estimating the unknown input vector:

- the mapping from the unknown input vector to the residual vector satisfies

$$H_{\bar{d}}(N) H_{\bar{d}}^T(N) = I,$$

when the unified solution is adopted. In this sense, the residual vector delivers an estimation for the unknown input vector $d(k)$, and

- the observer-based residual generator with

$$L_o(k) = \left(A(k)P(k)C^T(k) + E_d(k)F_d^T(k)\right) V_o^2(k)$$

can be equivalently written as

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_8

$$\begin{aligned}\hat{x}(k+1) &= A(k)\hat{x}(k) + B(k)u(k) + L_1(k)r(k) + E_d\hat{d}(k),\\ \hat{d}(k) &= F_d^T(k)V_o^2(k)r(k), L_1(k) = A(k)P(k)C^T(k)V_o^2(k),\end{aligned}$$

in which $\hat{d}(k)$ can be interpreted as an estimation for the unknown input vector.

For our purpose, we will first introduce a least squares (LS) estimation scheme for systems with deterministic unknown inputs, which was intensively studied a couple of decades ago, but did not receive researcher's attention for an application for the fault estimation purpose. We will compare this LS estimation scheme with the application of the unified solution for fault estimation and discuss about their advantages and disadvantages.

8.1 Regularised Least Squares Estimation

We first introduce the so-called regularised least squares (RLS) estimation method for estimating x in a static process modelled by

$$y = Hx + v \tag{8.3}$$

with $x \in \mathcal{R}^n$ and measurement vector $y \in \mathcal{R}^m$. v is an unknown vector. The estimated x should solve the optimisation problem

$$\min_x J = \min_x \left(\|y - Hx\|_W^2 + \|x\|_{\Sigma^{-1}}^2 \right). \tag{8.4}$$

Here, $W > 0$, $\Sigma > 0$ are (given) weighting matrices. By a straightforward computation, we have the solution

$$\begin{aligned}&\frac{dJ}{dx} = 0 \Longrightarrow \Sigma^{-1}x = H^T W(y - Hx) \Longrightarrow\\ &\quad \hat{x} = \arg\min_x J = \left(\Sigma^{-1} + H^T WH\right)^{-1} H^T Wy.\end{aligned} \tag{8.5}$$

It is of interest to notice that if $\Sigma^{-1} \to 0$,

$$\hat{x} = \left(H^T WH\right)^{-1} H^T Wy$$

is a regular LS solution. We now extend the cost function to

$$J = \|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2, \tag{8.6}$$

where a is an estimate for x based on *a priori* knowledge. A (very) large Σ means that we are less confident with the estimate a . The solution for minimising (8.6) is given by

$$\hat{x} = \left(\Sigma^{-1} + H^T W H\right)^{-1} \left(H^T W y + \Sigma^{-1} a\right). \tag{8.7}$$

Since

$$\left(\Sigma^{-1} + H^T W H\right)^{-1} = \Sigma - \Sigma H^T \left(W^{-1} + H \Sigma H^T\right)^{-1} H \Sigma,$$

solution (8.7) can be further written into

$$\hat{x} = a + L\left(y - Ha\right), L = \Sigma H^T \left(W^{-1} + H \Sigma H^T\right)^{-1}. \tag{8.8}$$

This form of estimation for x is similar to a recursive LS algorithm and can be interpreted as a correction of a (as a pre-estimate of x) by the difference $(y - Ha)$, which acts like a residual vector.

For our purpose, we would like to mention some facts with the RLS algorithm, which are useful for our subsequent work:

- By means of y and $\hat{x}$, v can be estimated in terms of

$$\hat{v} = y - H\hat{x} = W^{-1} \left(W^{-1} + H \Sigma H^T\right)^{-1} (y - Ha). \tag{8.9}$$

- It holds

$$\begin{aligned} J &= \|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 \\ &= \left\|y - H\hat{x}\right\|_W^2 + \left\|\hat{x} - a\right\|_{\Sigma^{-1}}^2 + \left\|x - \hat{x}\right\|_{\Sigma^{-1}}^2 + \left\|v - \hat{v}\right\|_W^2 \qquad (8.10) \\ &= \left\|y - H\hat{x}\right\|_W^2 + \left\|\hat{x} - a\right\|_{\Sigma^{-1}}^2 + \left\|x - \hat{x}\right\|_{P^{-1}}^2, \\ P^{-1} &= H^T W H + \Sigma^{-1}. \qquad (8.11) \end{aligned}$$

- Variable vectors $\hat{x}$, $\hat{v}$ are the solution of the optimisation problem

$$\begin{aligned} &\min_{x,v} J,\ J = \|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 = \|v\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 \\ &\quad \text{s.t. } y = Hx + v \end{aligned}$$

with the optimal value of the cost function

$$\min_{x,v} J = \left\|y - H\hat{x}\right\|_W^2 + \left\|\hat{x} - a\right\|_{\Sigma^{-1}}^2. \tag{8.12}$$

It is clear that (8.12) is the result of (8.10). In fact, relation (8.10) reveals that

$$\min_x J = \min_x \left(\|y - Hx\|_W^2 + \|x\|_{\Sigma^{-1}}^2\right)$$

is equivalent with

$$\begin{aligned} \min_{x,v} J &= \min_{x,v} \left(\|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 \right) \\ &\text{s.t. } y = Hx + v. \end{aligned} \tag{8.13}$$

In order to prove (8.10), we do the following calculations: Let

$$e_v = v - \hat{v}, e_x = x - \hat{x}.$$

Then,

$$\begin{aligned} &\|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 = \left\| \hat{v} + e_v \right\|_W^2 + \left\| \hat{x} - a + e_x \right\|_{\Sigma^{-1}}^2 \\ &= \left\| \hat{v} \right\|_W^2 + \left\| \hat{x} - a \right\|_{\Sigma^{-1}}^2 + \|e_v\|_W^2 + \|e_x\|_{\Sigma^{-1}}^2 + 2\left(\hat{v}^T W e_v + \left(\hat{x} - a \right)^T \Sigma^{-1} e_x \right). \end{aligned}$$

By noting that

$$\begin{gathered} \hat{v}^T W e_v = (y - Ha)^T \left(W^{-1} + H \Sigma H^T \right)^{-1} H \left(-e_x \right), \\ \left(\hat{x} - a \right)^T \Sigma^{-1} e_x = (y - Ha)^T \left(W^{-1} + H \Sigma H^T \right)^{-1} H e_x, \\ \|e_v\|_W^2 = e_x^T H^T W H e_x, \; P^{-1} = H^T W H + \Sigma^{-1} \Longrightarrow \\ \|e_v\|_W^2 + \|e_x\|_{\Sigma^{-1}}^2 = \|e_x\|_{P^{-1}}^2, \end{gathered}$$

equation (8.10) is proved. It is remarkable that

$$\hat{v}^T W e_v + \left(\hat{x} - a \right)^T \Sigma^{-1} e_x = \left[\hat{v}^T \; \left(\hat{x} - a \right)^T \right] \begin{bmatrix} W & 0 \\ 0 & \Sigma^{-1} \end{bmatrix} \begin{bmatrix} e_v \\ e_x \end{bmatrix} = 0$$

means that vectors

$$\begin{bmatrix} \hat{v} \\ \hat{x} - a \end{bmatrix} \text{ and } \begin{bmatrix} e_v \\ e_x \end{bmatrix}$$

are orthogonal. Since e_v, e_x are the estimation errors of v, x, $\hat{v}$ and $\hat{x} - a$ are understood as the LS estimates for v and $x - a$ (and so x), respectively.

Example 8.1 *As an extension of the optimisation problem (8.13), we consider*

$$\min_{x,v} J = \min_{x,v} \left(\|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 \right) \tag{8.14}$$

$$s.t. \; y = Hx + Fv, rank(F) = m. \tag{8.15}$$

It follows from the previous results that

$$\begin{gathered}\|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2 = \left\|\hat{\bar{v}} + e_{\bar{v}}\right\|_W^2 + \left\|\hat{x} - a + e_x\right\|_{\Sigma^{-1}}^2 \\ = \left\|\hat{\bar{v}}\right\|_W^2 + \left\|\hat{x} - a\right\|_{\Sigma^{-1}}^2 + \|e_{\bar{v}}\|_W^2 + \|e_x\|_{\Sigma^{-1}}^2 , \\ \hat{\bar{v}} = y - H\hat{x} = W^{-1}\left(W^{-1} + H\Sigma H^T\right)^{-1}(y - Ha), \\ e_{\bar{v}} = Fv - \hat{\bar{v}}, e_x = x - \hat{x}.\end{gathered}$$

Recall that

$$\hat{v} = F^T\left(FF^T\right)^{-1}\hat{\bar{v}}$$

is an LS estimate of v satisfying

$$F\hat{v} = \hat{\bar{v}}.$$

As a result, for

$$\begin{gathered}\hat{x} = a + L(y - Ha), L = \Sigma H^T\left(W^{-1} + H\Sigma H^T\right)^{-1}, \\ \hat{v} = F^T\left(FF^T\right)^{-1}W^{-1}\left(W^{-1} + H\Sigma H^T\right)^{-1}(y - Ha),\end{gathered}$$

it holds

$$\begin{gathered}\min_{x,v}\left(\|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2\right) = \left\|\hat{\bar{v}}\right\|_W^2 + \left\|\hat{x} - a\right\|_{\Sigma^{-1}}^2 = \|e_x\|_{P^{-1}}^2 , \\ P^{-1} = H^T W H + \Sigma^{-1}.\end{gathered}$$

That is

$$\left\{\hat{x}, \hat{v}\right\} = \arg\min_{x,v}\left(\|y - Hx\|_W^2 + \|x - a\|_{\Sigma^{-1}}^2\right).$$

Note that for $W = \left(FF^T\right)^{-1}$,

$$\hat{v} = F^T\left(FF^T + H\Sigma H^T\right)^{-1}(y - Ha).$$

RLS algorithm is widely applied in the machine learning technique and plays an important role in our subsequent investigation.

8.2 Least Squares Observer and Sensor Fault Estimation

8.2.1 Problem Formulation

We now formulate the fault estimation as an optimisation problem: given measurement data $y(0), \cdots, y(k), \cdots, y(N)$, find $x(k), f(k), k = 0, \cdots, N$, such that the cost function

$$J_N = \frac{1}{2}\begin{pmatrix} \|x(0) - x_o\|^2_{P^{-1}(0)} + \|y(k) - C(k)x(k)\|^2_{W_1(k),2,[0,N]} \\ + \|x(k+1) - A(k)x(k)\|^2_{W_2(k),2,[0,N-1]} \end{pmatrix} \tag{8.16}$$

$$\text{s.t. } x(k+1) = A(k)x(k) + E(k)f(k), \tag{8.17}$$

$$y(k) = C(k)x(k) + F(k)f(k) \tag{8.18}$$

is minimised, where

$$W_1(k) = \left(F(k)F^T(k)\right)^{-1}, W_2(k) \in \mathcal{R}^{n\times n}, rank(W_2(k)) = n,$$

are (symmetric) weighting matrices, and it is assumed that

$$rank\,(F(k)) = m, F(k) \in \mathcal{R}^{m\times k_f}, \tag{8.19}$$

$P(0) > 0$ is given and x_o is an initial estimation for $x(0)$ based on *a priori* knowledge.

We call the solution of the above optimisation problem least squares (LS) estimations for $x(k)$, $f(k)$ and the associated dynamic system (estimator) least squares (LS) observer.

Remark 8.1 *To simplify our work, we omit the dynamics with respect to the process input $u(k)$. For linear systems without parameter uncertainties, this way of handling the estimation problem leads to no loss of generality.*

Remark 8.2 *Since the fault vector $f(k)$ directly affects the measurement vector $y(k)$, it is called sensor fault.*

8.2.2 Solution Algorithm

In order to gain deep insight into the solution, we are going to derive the solution and the associated algorithm step by step.

We begin with $N = 0$. In this case, the optimisation problem (8.16) can be reformulated as

$$\min_{x(0),f(0)} J_0, \tag{8.20}$$

$$J_0 = \frac{1}{2}\|x(0) - x_o\|^2_{P^{-1}(0)} + \frac{1}{2}\|y(0) - C(0)x(0)\|^2_{\left(F(0)F^T(0)\right)^{-1}},$$

$$\text{s.t. } y(0) = C(0)x(0) + F(0)f(0).$$

It follows from the RLS algorithm and the associated discussion that

$$\hat{x}(0) = x_o + L(0)\left(y(0) - C(0)x_o\right), L(0) = P(0)C^T(0)R^{-1}(0), \quad (8.21)$$

$$\hat{f}(0) = F^T(0)R^{-1}(0)\left(y(0) - C(0)x_o\right), \quad (8.22)$$

$$R(0) = F(0)F^T(0) + C(0)P(0)C^T(0). \quad (8.23)$$

For our purpose, we denote $\hat{x}(0)$ by $\hat{x}(0\,|0\,)$. Moreover, the cost function

$$J_0 = \frac{1}{2}\left\|x(0) - x_o\right\|^2_{P^{-1}(0)} + \frac{1}{2}\left\|y(0) - C(0)x(0)\right\|^2_{\left(F(0)F^T(0)\right)^{-1}}$$

can be written as

$$J_0 = J(0) + \frac{1}{2}\left\|F(0)\left(f(0) - \hat{f}(0)\right)\right\|^2_{\left(F(0)F^T(0)\right)^{-1}} + \frac{1}{2}\left\|x(0) - \hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0)},$$
$$(8.24)$$
$$J(0) = \frac{1}{2}\left\|\hat{x}(0\,|0\,) - x_o\right\|^2_{P^{-1}(0)} + \frac{1}{2}\left\|F(0)\hat{f}(0)\right\|^2_{\left(F(0)F^T(0)\right)^{-1}}.$$

Notice that

$$\left\|F(0)\left(f(0) - \hat{f}(0)\right)\right\|^2_{\left(F(0)F^T(0)\right)^{-1}} = \left\|x(0) - \hat{x}(0\,|0\,)\right\|^2_{C^T(0)\left(F(0)F^T(0)\right)^{-1}C(0)},$$

and let

$$P^{-1}(0\,|0\,) = P^{-1}(0) + C^T(0)\left(F(0)F^T(0)\right)^{-1}C(0).$$

It turns out

$$J_0 = J(0) + \frac{1}{2}\left\|F(0)\left(f(0) - \hat{f}(0)\right)\right\|^2_{\left(F(0)F^T(0)\right)^{-1}} + \frac{1}{2}\left\|x(0) - \hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0)}$$
$$= J(0) + \frac{1}{2}\left\|x(0) - \hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0|0\,)}. \quad (8.25)$$

Next, we study the optimisation problem for $N = 1$, which, following our result given in (8.25), can be formulated as

$$\min_{x(k), f(k), k=0,1} J_1,$$
$$J_1 = J(0) + \frac{1}{2}\left\|x(0) - \hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0|0\,)} + \frac{1}{2}\left\|x(1) - A(0)x(0)\right\|^2_{W_2(0)}$$
$$+\frac{1}{2}\left\|y(1) - C(1)x(1)\right\|^2_{\left(F(1)F^T(1)\right)^{-1}},$$
$$\text{s.t. } x(1) = A(0)x(0) + E(0)f(0),$$
$$y(1) = C(1)x(1) + F(1)f(1).$$

Noting that $J(0)$ is a constant value, this optimisation problem is equivalent with

$$\min_{x(k),f(k),k=0,1} \bar{J}_1, \tag{8.26}$$

$$\begin{aligned}\bar{J}_1 = \frac{1}{2}\left\|x(0)-\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0|0\,)} + \frac{1}{2}\left\|x(1)-A(0)x(0)\right\|^2_{W_2(0)}\\ +\frac{1}{2}\left\|y(1)-C(1)x(1)\right\|^2_{\left(F(1)F^T(1)\right)^{-1}},\\ \text{s.t. } x(1) = A(0)x(0)+E(0)f(0),\\ y(1) = C(1)x(1)+F(1)f(1).\end{aligned}$$

We solve this problem in two steps. It is evident that the estimate for $x(0)$ does not explicitly depend on $y(1)$. Therefore, in the first step, we consider the optimisation problem

$$\begin{aligned}\min_{x(0),f(0)}\left(\frac{1}{2}\left\|x(0)-\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0|0\,)} + \frac{1}{2}\left\|x(1)-A(0)x(0)\right\|^2_{W_2(0)}\right)\\ \text{s.t. } x(1) = A(0)x(0)+E(0)f(0),\end{aligned} \tag{8.27}$$

and express the solution for $x(0)$, denoted by $\hat{x}(0\,|1\,)$, as a function of $x(1)$. Note that the optimisation problem (8.27) has the identical form like (8.20) and can thus be solved using the RLS algorithm. It turns out

$$\begin{aligned}\hat{x}(0\,|1\,) = \hat{x}(0\,|0\,) + P(0\,|0\,)A^T(0)Q^{-1}\left(x(1)-A(0)\hat{x}(0\,|0\,)\right),\\ Q = W_2^{-1}(0) + A(0)P(0\,|0\,)A^T(0),\end{aligned}$$

which yields

$$\begin{aligned}&\min_{x(0),f(0)}\left(\frac{1}{2}\left\|x(0)-\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0|0\,)} + \frac{1}{2}\left\|x(1)-A(0)x(0)\right\|^2_{W_2(0)}\right)\\ &= \frac{1}{2}\left\|P(0\,|0\,)A^T(0)Q^{-1}\left(x(1)-A(0)\hat{x}(0\,|0\,)\right)\right\|^2_{P^{-1}(0|0\,)}\\ &\quad+\frac{1}{2}\left\|x(1)-A(0)\hat{x}(0\,|1\,)\right\|^2_{W_2(0)}.\end{aligned}$$

Recall

$$\begin{aligned}x(1)-A(0)\hat{x}(0\,|1\,) &= \left(I-A(0)P(0\,|0\,)A^T(0)Q^{-1}\right)\left(x(1)-A(0)\hat{x}(0\,|0\,)\right)\\ &= W_2^{-1}(0)Q^{-1}\left(x(1)-A(0)\hat{x}(0\,|0\,)\right).\end{aligned}$$

It holds

$$\begin{aligned}
&\left\| P(0\,|0\,)A^T(0)Q^{-1}\left(x(1)-A(0)\hat{x}(0\,|0\,)\right)\right\|^2_{P^{-1}(0|0\,)} + \left\|x(1)-A(0)\hat{x}(0\,|1\,)\right\|^2_{W_2(0)} \\
&\qquad = \left\|x(1)-A(0)\hat{x}(0\,|0\,)\right\|_{P^{-1}(1|0\,)}, \\
&\qquad P(1\,|0\,) := Q = W_2^{-1}(0) + A(0)P(0\,|0\,)A^T(0).
\end{aligned}$$

Hence, after the first step, we have

$$\begin{aligned}
&\min_{x(k),f(k),k=0,1} \bar{J}_1 \\
&= \min_{x(1),f(1)} \frac{1}{2}\left(\left\|x(1)-\hat{x}(1\,|0\,)\right\|_{P^{-1}(1|0\,)} + \|y(1)-C(1)x(1)\|^2_{\left(F(1)F^T(1)\right)^{-1}}\right) \\
&\text{s.t. } y(1) = C(1)x(1)+F(1)f(1) \text{ with} \\
&\hat{x}(1\,|0\,) = A(0)\hat{x}(0\,|0\,).
\end{aligned}$$

This is the same optimisation problem like (8.20) for $N=0$, and thus is solved by

$$\hat{x}(1\,|1\,) = \hat{x}(1\,|0\,) + L(1)\left(y(1)-C(1)\hat{x}(1\,|0\,)\right), \tag{8.28}$$

$$\hat{f}(1) = F^T(1)R^{-1}(1)\left(y(1)-C(1)\hat{x}(1\,|0\,)\right), \tag{8.29}$$

$$L(1) = P(1\,|0\,)C^T(1)R^{-1}(1), \tag{8.30}$$

$$R(1) = F(1)F^T(1) + C(1)P(1\,|0\,)C^T(1). \tag{8.31}$$

Moreover, it holds

$$\begin{aligned}
\bar{J}_1 &= \frac{1}{2}\left\|x(1)-\hat{x}(1\,|0\,)\right\|^2_{P^{-1}(1|0\,)} + \frac{1}{2}\|F(1)f(1)\|^2_{\left(F(1)F^T(1)\right)^{-1}} \\
&= J(1) + \frac{1}{2}\left\|x(1)-\hat{x}(1\,|1\,)\right\|^2_{P^{-1}(1|1\,)}, \qquad (8.32)
\end{aligned}$$

$$\begin{aligned}
P^{-1}(1\,|1\,) &= P^{-1}(1\,|0\,) + C^T(1)\left(F(1)F^T(1)\right)^{-1}C(1), \\
\hat{x}(2\,|1\,) &= A(1)\hat{x}(1\,|1\,), \\
J(1) &= \frac{1}{2}\left\|\hat{x}(1\,|1\,)-\hat{x}(1\,|0\,)\right\|^2_{P^{-1}(1|0\,)} + \frac{1}{2}\left\|F(1)\hat{f}(1)\right\|^2_{\left(F(1)F^T(1)\right)^{-1}},
\end{aligned}$$

$$J_1 = J(0) + J(1) + \frac{1}{2}\left\|x(1)-\hat{x}(1\,|1\,)\right\|^2_{P^{-1}(1|1\,)}. \tag{8.33}$$

With the aid of the above study, we are now in a position to give and prove the solution for the original optimisation problem (8.16). To this end, we assume that for $N=i$,

$$J_i = \frac{1}{2}\begin{pmatrix} \|x(0)-x_o\|^2_{P^{-1}(0)} + \|y(k)-C(k)x(k)\|^2_{W_1(k),2,[0,i]} \\ + \|x(k+1)-A(k)x(k)\|^2_{W_2(k),2,[0,i-1]} \end{pmatrix}$$

$$= \sum_{k=0}^{i} J(k) + \frac{1}{2}\left\|x(i)-\hat{x}(i\,|i\,)\right\|^2_{P^{-1}(i|i\,)},$$

$$J(k) = \frac{1}{2}\left\|\hat{x}(k\,|k\,)-\hat{x}(k\,|k-1\,)\right\|^2_{P^{-1}(k|k-1\,)} + \frac{1}{2}\left\|F(k)\hat{f}(k)\right\|^2_{W_1(k)},$$

$$\hat{x}(i\,|i\,) = \hat{x}(i\,|i-1\,) + L(i)\left(y(i)-C(i)\hat{x}(i\,|i-1\,)\right), \hat{x}(0\,|{-1}\,) := x_o, \quad (8.34)$$

$$\hat{x}(i+1\,|i\,) = A(i)\hat{x}(i\,|i\,), \quad (8.35)$$

$$\hat{f}(i) = F^T(i)R^{-1}(i)\left(y(i)-C(i)\hat{x}(i\,|i-1\,)\right), \quad (8.36)$$

$$L(i) = P(i\,|i-1\,)C^T(i)R^{-1}(i), \quad (8.37)$$

$$R(i) = F(i)F^T(i) + C(i)P(i\,|i-1\,)C^T(i), \quad (8.38)$$

$$P^{-1}(i\,|i\,) = P^{-1}(i\,|i-1\,) + C^T(i)\left(F(i)F^T(i)\right)^{-1}C(i),\ P(0\,|{-1}\,) = P(0), \quad (8.39)$$

$$P(i\,|i-1\,) = W_2^{-1}(i-1) + A(i-1)P(i-1\,|i-1\,)A^T(i-1), \quad (8.40)$$

and derive the solution for $N = i + 1$. It is clear that

$$J_{i+1} = \frac{1}{2}\begin{pmatrix} \|x(0)-x_o\|^2_{P^{-1}(0)} + \|y(k)-C(k)x(k)\|^2_{W_1(k),2,[0,i+1]} \\ + \|x(k+1)-A(k)x(k)\|^2_{W_2(k),2,[0,i]} \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} \|x(0)-x_o\|^2_{P^{-1}(0)} + \|y(k)-C(k)x(k)\|^2_{W_1(k),2,[0,i]} \\ + \|x(k+1)-A(k)x(k)\|^2_{W_2(k),2,[0,i-1]} \\ + \|y(i+1)-C(i+1)x(i+1)\|^2_{W_1(i+1)} \\ + \|x(i+1)-A(i)x(i)\|^2_{W_2(i)} \end{pmatrix}$$

$$= \sum_{k=0}^{i} J(k) + \frac{1}{2}\begin{pmatrix} \left\|x(i)-\hat{x}(i\,|i\,)\right\|^2_{P^{-1}(i|i\,)} + \|x(i+1)-A(i)x(i)\|^2_{W_2(i)} \\ + \|y(i+1)-C(i+1)x(i+1)\|^2_{W_1(i+1)} \end{pmatrix}.$$

Recall that

$$\left\|x(i)-\hat{x}(i\,|i\,)\right\|^2_{P^{-1}(i|i\,)} + \|x(i+1)-A(i)x(i)\|^2_{W_2(i)}$$

can be written as

$$\left\|P(i\,|i\,)A^T(i)P^{-1}(i+1\,|i\,)\left(x(i+1)-A(i)\hat{x}(i\,|i\,)\right)\right\|^2_{P^{-1}(i|i\,)}$$
$$+ \left\|x(i+1)-A(i)\hat{x}(i)\right\|^2_{W_2(i)} = \left\|x(i+1)-A(i)\hat{x}(i\,|i\,)\right\|_{P^{-1}(i+1|i\,)},$$
$$P(i+1\,|i\,) = W_2^{-1}(i) + A(i)P(i\,|i\,)A^T(i).$$

It yields

$$J_{i+1}=\sum_{k=0}^{i}J(k)+\frac{1}{2}\left(\begin{array}{c}\left\|x(i+1)-A(i)\hat{x}(i\,|i\,)\right\|_{P^{-1}(i+1|i\,)}^{2}\\+\left\|y(i+1)-C(i+1)x(i+1)\right\|_{W_1(i+1)}^{2}\end{array}\right),$$

which can be, using the RLS algorithm, brought into

$$\begin{aligned}J_{i+1}&=\sum_{k=0}^{i}J(k)+J(i+1)+\frac{1}{2}\left\|x(i+1)-\hat{x}(i+1\,|i+1\,)\right\|_{P^{-1}(i+1|i+1\,)}^{2},\\J(i+1)&=\frac{1}{2}\left(\begin{array}{c}\left\|\hat{x}(i+1\,|i+1\,)-\hat{x}(i+1\,|i\,)\right\|_{P^{-1}(i+1|i\,)}^{2}\\+\left\|F(i+1)\hat{f}(i+1)\right\|_{\left(F(i+1)F^{T}(i+1)\right)^{-1}}^{2}\end{array}\right),\\\hat{x}(i+1\,|i+1\,)&=\hat{x}(i+1\,|i\,)+L(i+1)\left(y(i+1)-C(i+1)\hat{x}(i+1\,|i\,)\right),\\\hat{x}(i+1\,|i\,)&=A(i)\hat{x}(i\,|i\,),\\\hat{f}(i+1)&=F^{T}(i+1)R^{-1}(i+1)\left(y(i+1)-C(i+1)\hat{x}(i+1\,|i\,)\right),\\L(i+1)&=P(i+1\,|i\,)C^{T}(i+1)R^{-1}(i+1),\\R(i+1)&=F(i+1)F^{T}(i+1)+C(i+1)P(i+1\,|i\,)C^{T}(i+1),\\P(i+1\,|i\,)&=W_{2}^{-1}(i)+A(i)P(i\,|i\,)A^{T}(i),\\P^{-1}(i+1\,|i+1\,)&=P^{-1}(i+1\,|i\,)+C^{T}(i+1)\left(F(i+1)F^{T}(i+1)\right)^{-1}C(i+1).\end{aligned}$$

By mathematical induction, we have proved the following theorem.

Theorem 8.1 *The solution of the optimisation problem (8.16)–(8.18) is given by (8.34)–(8.40) for $i=k$. Moreover,*

$$\begin{aligned}\min_{x(k),f(k),k=0,\cdots,N}J_{N}&=\frac{1}{2}\left(\begin{array}{c}\left\|\hat{x}(k\,|k\,)-\hat{x}(k\,|k-1\,)\right\|_{P^{-1}(k|k-1\,),2,[0,N]}^{2}+\\\left\|y(k)-C(k)\hat{x}(k\,|k\,)\right\|_{\left(F(k)F^{T}(k)\right)^{-1},2,[0,N]}^{2}\end{array}\right)\\&=\frac{1}{2}\left\|y(k)-C(k)\hat{x}(k\,|k-1\,)\right\|_{R^{-1}(k),2,[0,N]}^{2}.\end{aligned}\tag{8.41}$$

Here, (8.41) follows from the equations

$$\begin{aligned}\hat{x}(k\,|k\,)-\hat{x}(k\,|k-1\,)&=L(k)\left(y(k)-C(k)\hat{x}(k\,|k-1\,)\right),\\y(k)-C(k)\hat{x}(k\,|k\,)&=F(k)F^{T}(k)R^{-1}(k)\left(y(k)-C(k)\hat{x}(k\,|k-1\,)\right),\\L(k)&=P(k\,|k-1\,)C^{T}(k)R^{-1}(k),\\R(k)&=F(k)F^{T}(k)+C(k)P(k\,|k-1\,)C^{T}(k).\end{aligned}$$

8.2.3 Discussions

We now discuss about the achieved solution and results from different aspects.

One-step ahead prediction

Although the focus of our previous study is on the estimation $\hat{x}(k\,|k)$, one-step prediction $\hat{x}(k\,|k-1)$ plays a central role, since both $\hat{x}(k\,|k)$, $\hat{f}(k)$ are driven by the residual vector $y(k) - C(k)\hat{x}(k\,|k-1)$. This motivates us to discuss about the one-step prediction issue.

It is obvious that

$$\hat{x}(k+1\,|k) = A(k)\hat{x}(k\,|k-1) + L\,(k+1\,|k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right), \quad (8.42)$$

$$L\,(k+1\,|k) = A(k)L(k) = A(k)P(k\,|k-1)C^T(k)R^{-1}(k). \quad (8.43)$$

Suppose that we are now interested in finding $x(k+1), f(k), k = 0, \cdots, N-1$, so that for the measurement data, $y(0), \cdots, y(k), \cdots, y(N-1)$, the cost function

$$J_{N|N-1} = \frac{1}{2}\begin{pmatrix} \|x(0) - x_o\|^2_{P^{-1}(0)} + \|y(k) - C(k)x(k)\|^2_{W_1(k),2,[0,N-1]} \\ +\|x(k+1) - A(k)x(k)\|^2_{W_2(k),2,[0,N-1]} \end{pmatrix}$$

is minimised subject to (8.17)–(8.18). Using the results achieved in the last sub-section, the following theorem can be proved, which provides us with a solution of the above optimisation problem.

Theorem 8.2 *Given* $x_o := \hat{x}(0\,|-1), y(0), \cdots, y(k), \cdots, y(N-1)$, *the solution to the minimisation of the cost function* $J_{N|N-1}$ *subject to (8.17)–(8.18) is given by*

$$\left\{\hat{x}(k+1\,|k), \hat{f}(k), k = 0, \cdots, N-1\right\} = \arg \min_{x(k+1), f(k), k=0,\cdots N-1} J_{N|N-1},$$

$$\min_{x(k+1), f(k), k=0,\cdots N-1} J_{N|N-1} = \frac{1}{2}\left\|y(k) - C(k)\hat{x}(k\,|k-1)\right\|^2_{R^{-1}(k),2,[0,N-1]}, \quad (8.44)$$

$$R(k) = F(k)F^T(k) + C(k)P(k\,|k-1)C^T(k).$$

Proof It follows from our discussion in the last sub-section that

$$J_{N|N-1} = \frac{1}{2}\begin{pmatrix} \left\|\hat{x}(k\,|k) - \hat{x}(k\,|k-1)\right\|^2_{P^{-1}(k|k-1),2,[0,N-1]} \\ +\left\|F(k)\hat{f}(k)\right\|^2_{\left(F(k)F^T(k)\right)^{-1},2,[0,N-1]} \end{pmatrix}$$
$$+\frac{1}{2}\left\|x(N) - \hat{x}(N\,|N-1)\right\|^2_{P^{-1}(N|N-1)}.$$

Recalling

$$\hat{x}(k\,|k) - \hat{x}(k\,|k-1) = L(k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right),$$
$$F(k)\hat{f}(k) = R^{-1}(k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right),$$

equation (8.44) is obviously true.

About the update of $P(k\,|k\,)$, $P(k+1\,|k\,)$

It is straightforward that the update of $P(k\,|k\,)$, $P(k+1\,|k\,)$ can be written in the recursive forms

$$\begin{gathered}
P(k\,|k\,) = \left(P^{-1}(k\,|k-1\,) + C^T(k)\left(F(k)F^T(k)\right)^{-1}C(k)\right)^{-1} \Longrightarrow \\
P(k+1\,|k+1\,) = W_2^{-1}(k) + A(k)P(k\,|k\,)A^T(k) \\
-L(k+1)R(k+1)L^T(k+1), \\
P(k+1\,|k\,) = W_2^{-1}(k) + A(k)P(k\,|k\,)A^T(k) \\
= W_2^{-1}(k) + A(k)P(k\,|k-1\,)A^T(k) - L(k+1\,|k\,)R(k)L^T(k+1\,|k\,), \\
L(k+1) = P(k+1\,|k\,)C^T(k+1)R^{-1}(k+1), L(k+1\,|k\,) = A(k)L(k), \\
R(k) = F(k)F^T(k) + C(k)P(k\,|k-1\,)C^T(k),
\end{gathered}$$

which are Riccati recursions and represent the dual form of the well-known LQ regulator. For an LTI detectable system with $N \to \infty$, $P(N\,|N-1\,)$ is the solution of an algebraic Riccati equation

$$\begin{gathered}
\lim_{N\to\infty} P(N\,|N-1\,) = P > 0, \\
APA^T - P - LRL^T + W_2^{-1} = 0, \\
L = APC^TR^{-1}, R = FF^T + CPC^T,
\end{gathered} \tag{8.45}$$

and moreover

$$\lim_{N\to\infty} P(N\,|N\,) = \bar{P}, \bar{P} = \left(P^{-1} + C^T\left(FF^T\right)^{-1}C\right)^{-1}.$$

Like the covariance matrix of the state estimation error in Kalman filter, $P(k+1\,|k\,)$ and $P(k\,|k\,)$ are also an indicator for the estimation performance, but with different interpretation, as will be discussed below.

About the cost function

Write the cost function as

$$\begin{gathered}
J_N = \frac{1}{2}\left(\|x(0) - x_o\|^2_{P^{-1}(0)} + \|f(k)\|^2_{W_f(k),2,[0,N]}\right), \\
W_f(k) = F^T(k)\left(F(k)F^T(k)\right)^{-1}F(k) + E^T(k)W_2(k)E(k),
\end{gathered}$$

whose optimum value is

$$J_N^* = \frac{1}{2} \left\| y(k) - C(k)\hat{x}(k\,|k-1) \right\|^2_{R^{-1}(k),2,[0,N]}.$$

Let

$$r(k) = R^{-1/2}(k) \left(y(k) - C(k)\hat{x}(k\,|k-1) \right)$$

be the (optimal) residual vector with a post-filter $R^{-1/2}(k)$. Since $J_N \geq J_N^*$, it holds

$$\|r(k)\|^2_{2,[0,N]} \leq \|x(0) - x_o\|^2_{P^{-1}(0)} + \|f(k)\|^2_{W_f(k),2,[0,N]}. \tag{8.46}$$

On the assumption of a good estimation of the initial value of $x(0)$ and selecting $W_2(k)$ so that

$$W_f(k) = F^T(k) \left(F(k)F^T(k) \right)^{-1} F(k) + E^T(k)W_2(k)E(k) = \gamma^{-1}I, \gamma > 0, \tag{8.47}$$

a lower bound $\|f(k)\|^2_{2,[0,N]}$ can be expressed in terms of the $l_{2,[0,N]}$ norm of the residual vector, that is

$$\|f(k)\|^2_{2,[0,N]} \geq \gamma \, \|r(k)\|^2_{2,[0,N]}.$$

Example 8.2 *In this example, we demonstrate the determination of the weighting matrices $W_1(k)$ and $W_2(k)$ aiming to ensure equation (8.47). To be specific, we are interested in the case $\gamma = 1$. It is evident that for $W_1(k) = \left(F(k)F^T(k) \right)^{-1}$,*

$$I - F^T(k) \left(F(k)F^T(k) \right)^{-1} F(k) \geq 0 \Longrightarrow$$
$$E^T(k)W_2(k)E(k) = I - F^T(k) \left(F(k)F^T(k) \right)^{-1} F(k) \geq 0.$$

That means, on the assumption of

$$rank(E(k)) = k_f,$$

the requirement that

$$rank\,(W_2(k)) = n$$

cannot be guaranteed. Now, let

$$W_1(k) = \eta \left(F(k)F^T(k) \right)^{-1}, 0 < \eta < 1, \tag{8.48}$$

which yields

$$I - F^T(k)W_1(k)F(k) = I - \eta F^T(k) \left(F(k)F^T(k) \right)^{-1} F(k) > 0.$$

Next, define

$$W_2(k) = \left[\left(E^-(k)\right)^T \ \left(E^\perp(k)\right)^T \right] \begin{bmatrix} I - F^T(k)W_1(k)F(k) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} E^-(k) \\ E^\perp(k) \end{bmatrix}, \tag{8.49}$$

$$rank \begin{bmatrix} E^-(k) \\ E^\perp(k) \end{bmatrix} = n, \begin{bmatrix} E^-(k) \\ E^\perp(k) \end{bmatrix} \in \mathcal{R}^{n\times n},$$

$$E^-(k)E(k) = I_{k_f \times k_f}, E^\perp(k)E(k) = 0.$$

It is clear that

$$E^T(k)W_2(k)E(k) = I - F^T(k)W_1(k)F(k), W_2(k) > 0.$$

In other words, it holds

$$\eta F^T(k)\left(F(k)F^T(k)\right)^{-1} F(k) + E^T(k)W_2(k)E(k) = I. \tag{8.50}$$

Recall that in the cost function J_N, $P^{-1}(0) = P^{-1}(0\,|{-1}\,)$ is an indicator for the confidence of the initial estimation x_o for $x(0)$. A smaller $P^{-1}(0)$ means a higher confidential degree. In general, it holds

$$\begin{aligned} J_N &= \frac{1}{2}\left(\|x(0) - x_o\|^2_{P^{-1}(0)} + \|f(k)\|^2_{W_f(k),2,[0,N]}\right) \\ &= \sum_{k=0}^{N-1} J(k) + \frac{1}{2}\left\|x(N) - \hat{x}(N\,|N-1\,)\right\|^2_{P^{-1}(k|k-1\,)}, \\ J(k) &= \frac{1}{2}\left\|\hat{x}(k\,|k\,) - \hat{x}(k\,|k-1\,)\right\|^2_{P^{-1}(k|k-1\,)} + \frac{1}{2}\left\|F(k)\hat{f}(k)\right\|^2_{\left(F(k)F^T(k)\right)^{-1}}. \end{aligned}$$

Hence, $P(k\,|k-1\,)$ indicates the confidential degree of the estimate $\hat{x}(k\,|k-1\,)$ for $x(k)$. Similarly, $P(k\,|k\,)$ can be interpreted as the confidence of the estimate $\hat{x}(k\,|k\,)$ for $x(k)$.

At the end of our discussion, we would like to point out that the optimisation problem (8.16)–(8.18) can be formulated in a recursive form as follows:

$$\begin{aligned} J_N^* &= \min_{x(k),f(k),k=0,\cdots N} J_N \\ &= \min_{x(N),f(N)} \begin{pmatrix} \min_{x(k),f(k),k=0,\cdots,N-1} J_{N-1} + \\ \frac{1}{2}\left\|x(N) - \hat{x}(N\,|N-1)\right\|^2_{P^{-1}(N|N-1\,)} + \|F(N)f(N)\|^2_{W_1(N)} \end{pmatrix} \\ &= \min_{x(N),f(N)} \begin{pmatrix} J_{N-1}^* + \frac{1}{2}\left\|x(N) - \hat{x}(N\,|N-1\,)\right\|^2_{P^{-1}(N|N-1\,)} \\ + \|F(N)f(N)\|^2_{W_1(N)} \end{pmatrix}. \end{aligned} \tag{8.51}$$

About the fault estimation

We have learnt that an (optimal) estimate for $f(k)$ using date up to $y(k)$ is delivered by

$$\hat{f}(k) = F^T(k)R^{-1}(k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right).$$

On the other hand, it has been demonstrated, by the proof of Theorem 8.1, that

$$\begin{gathered}\left\|x(k+1) - \hat{x}(k+1\,|k)\right\|^2_{P^{-1}(k+1|k)} \\ = \left\|P(k\,|k)A^T(k)P^{-1}(k+1\,|k)\left(x(k+1) - A(k)\hat{x}(k\,|k)\right)\right\|^2_{P^{-1}(k|k)} \\ + \left\|W_2^{-1}(k)P^{-1}(k+1\,|k)\left(x(k+1) - A(k)\hat{x}(k\,|k)\right)\right\|^2_{W_2(k)}, \\ P(k+1\,|k) = W_2^{-1}(k) + A(k)P(k\,|k)A^T(k),\end{gathered}$$

in which

$$\begin{gathered}W_2^{-1}(k)P^{-1}(k+1\,|k)\left(x(k+1) - A(k)\hat{x}(k\,|k)\right) = x(k+1) - A(k)\hat{x}(k\,|k+1), \\ \hat{x}(k\,|k+1) = \hat{x}(k\,|k) + P(k\,|k)A^T(k)P^{-1}(k+1\,|k)\left(x(k+1) - A(k)\hat{x}(k\,|k)\right).\end{gathered}$$

The vector $\hat{x}(k\,|k+1)$ represents an estimate for $x(k)$ in terms of $\hat{x}(k\,|k)$ and $x(k+1)$. Recall that, due to the system dynamic constraint,

$$x(k+1) = A(k)x(k) + E(k)f(k),$$

the term $x(k+1) - A(k)\hat{x}(k\,|k+1)$ gives an estimate of

$$f_E(k) = E(k)f(k).$$

In this regard, substituting $x(k+1)$ by its estimate $\hat{x}(k+1\,|k+1)$ leads to an estimate of $f_E(k)$ using the data up to $k+1$,

$$\begin{aligned}&\hat{f}_E(k\,|k+1) = \hat{x}(k+1\,|k+1) - A(k)\hat{x}(k\,|k+1) \qquad (8.52)\\ &= W_2^{-1}(k)P^{-1}(k+1\,|k)\left(\hat{x}(k+1\,|k+1) - A(k)\hat{x}(k\,|k)\right) \\ &= W_2^{-1}(k)C^T(k+1)R^{-1}(k+1)\left(y(k+1) - C(k+1)\hat{x}(k+1\,|k)\right).\end{aligned}$$

As a result, on the assumption

$$rank(E(k)) = k_f,$$

we have an estimate for $f(k)$ (using the data up to $k+1$)

$$\begin{aligned}\hat{f}(k\,|k+1) &= \left(E^T(k)E(k)\right)^{-1} E^T(k)W_2^{-1}(k)C^T(k+1)R^{-1}(k+1)r(k+1),\\ r(k+1) &= y(k+1) - C(k+1)\hat{x}(k+1\,|k).\end{aligned}$$

Summary of the estimation algorithms

Below is a summary of the estimation algorithms achieved based on the above discussions.

Algorithm 8.1 *One-step ahead prediction* $\hat{x}(k+1\,|k)$:

$$\begin{aligned}\hat{x}(k+1\,|k) &= A(k)\hat{x}(k\,|k-1) + L\,(k+1\,|k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right),\\ L\,(k+1\,|k) &= A(k)P(k\,|k-1)C^T(k)R^{-1}(k),\\ P(k+1\,|k) &= A(k)P(k\,|k-1)A^T(k) - L(k+1\,|k)R(k)L^T(k+1\,|k)\\ &\quad +W_2^{-1}(k),\\ R(k) &= F(k)F^T(k) + C(k)P(k\,|k-1)C^T(k).\end{aligned}$$

Algorithm 8.2 *Estimation* $\hat{x}(k+1\,|k+1)$:

$$\begin{aligned}\hat{x}(k+1\,|k+1) &= A(k)\hat{x}(k\,|k) + L\,(k+1)\begin{pmatrix} y(k+1)\\ -C(k+1)A(k)\hat{x}(k\,|k)\end{pmatrix},\\ L(k+1) &= \left(W_2^{-1}(k) + A(k)P(k\,|k)A^T(k)\right)C^T(k+1)R^{-1}(k+1),\\ P(k+1\,|k+1) &= A(k)P(k\,|k)A^T(k) - L(k+1)R(k+1)L^T(k+1)\\ &\quad +W_2^{-1}(k),\\ R(k+1) &= C(k+1)\left(W_2^{-1}(k) + A(k)P(k\,|k)A^T(k)\right)C^T(k+1)\\ &\quad +F(k+1)F^T(k+1).\end{aligned}$$

Note that due to (8.52), the estimate $\hat{x}(k+1\,|k+1)$ *can be also equivalently written as*

$$\begin{aligned}\hat{x}(k+1\,|k+1) &= A(k)\hat{x}(k\,|k) + \hat{f}_E(k\,|k+1)\\ &\quad +L_1(k+1)\left(y(k+1) - C(k+1)A(k)\hat{x}(k\,|k)\right),\\ L_1(k+1) &= A(k)P(k\,|k)A^T(k)C^T(k+1)R^{-1}(k+1).\end{aligned}$$

Algorithm 8.3 *Fault estimation:*

$$\begin{aligned}\hat{f}(k) &= F^T(k)R^{-1}(k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right),\\ \hat{f}(k\,|k+1) &= \left(E^T(k)E(k)\right)^{-1} E^T(k)W_2^{-1}(k)C^T(k+1)R^{-1}(k+1)r(k+1),\\ r(k+1) &= y(k+1) - C(k+1)\hat{x}(k+1\,|k).\end{aligned}$$

At the end of our discussion, we would like to mention that there are different ways to solve the optimisation problem (8.16)–(8.18). In Chap. 20, we will introduce an alternative solution. Moreover, it is remarkable that the solution could also be expressed in a form, which is different from our solution expressed in terms of $\hat{x}(k+1\,|k)$, $\hat{x}(k\,|k)$ and $\hat{f}(k)$.

8.3 Fault Estimation: Least Squares Observer VS. Unified Solution

As mentioned at the beginning of this chapter, the unified solution delivers an estimate for the unknown input vector and can be thus applied for estimating the fault vector as well. This motivates us to compare the estimation performance of the LS observer introduced in the last section and the unified solution. To this end, we first summarise the application of the unified solution for fault estimation.

8.3.1 Sensor Fault Estimation Using the Unified Solution

Consider the LDTV system model (8.1)–(8.2) and recall the unified solution presented in Chap. 7 with the unknown (fault) vector $f(k)$:

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + L(k)\left(y(k) - C(k)\hat{x}(k)\right), &(8.53)\\ L(k) &= \left(A(k)P(k)C^T(k) + E(k)F^T(k)\right)R^{-1}(k),\\ R(k) &= F(k)F^T(k) + C(k)P(k)C^T(k),\\ P(k+1) &= A(k)P(k)A^T(k) + E(k)E^T(k) - L(k)R(k)L^T(k). &(8.54)\end{aligned}$$

For our purpose, (8.53) is equivalently written into

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + E(k)\hat{f}(k) + L_1(k)\left(y(k) - C(k)\hat{x}(k)\right), &(8.55)\\ \hat{f}(k) &:= F^T(k)R^{-1}(k)\left(y(k) - C(k)\hat{x}(k)\right), &(8.56)\\ L_1(k) &= A(k)P(k)C^T(k)R^{-1}(k), &(8.57)\\ L(k)\left(y(k) - C(k)\hat{x}(k)\right) &= L_1(k)\left(y(k) - C(k)\hat{x}(k)\right) + E(k)\hat{f}(k),\end{aligned}$$

in which an estimate for $f(k)$, $\hat{f}(k)$, is introduced. We are interested in analysis of the estimation performance of $\hat{f}(k)$.

First of all, from (8.53) it can be clearly seen that $\hat{x}(k)$ is an one-step ahead prediction of $x(k)$. Moreover, since

$$F(k)F^T(k) \leq R(k) \Longrightarrow R^{-1/2}(k)F(k)F^T(k)R^{-1/2}(k) \leq I,$$

it is clear that

$$\begin{gathered}
\hat{f}^T(k)\hat{f}(k) = r^T(k)R^{-1/2}(k)F(k)F^T(k)R^{-1/2}(k)r(k) \leq r^T(k)r(k) \\
\Longrightarrow \left\| \hat{f}(k) \right\|_{2,[0,N]}^2 \leq \| r(k) \|_{2,[0,N]}^2 = \left\| y(k) - C(k)\hat{x}(k) \right\|_{R^{-1}(k),2,[0,N]}^2, \\
r(k) = R^{-1/2}(k)\left(y(k) - C(k)\hat{x}(k)\right).
\end{gathered}$$

It follows from the results achieved in Chap. 7 on the unified solution that

$$\| r(k) \|_{2,[0,N]}^2 \leq \| f(k) \|_{2,[0,N]}^2 + \| e(0) \|^2,$$

with $e(0)$ as the estimation error of the initial state vector. On the assumption that faulty operations will first occur after a long normal operational period, it is reasonable to assume $e(0) \simeq 0$. Thus,

$$\left\| \hat{f}(k) \right\|_{2,[0,N]}^2 \leq \| f(k) \|_{2,[0,N]}^2.$$

8.3.2 Comparison Study

In order to compare the unified solution and the LS observer scheme, we add sub-index "*US*" to the relevant variables and parameters of the unified solutions and "*LS*" to the ones of the LS observer. We have

- one-step ahead prediction of $x(k)$ in comparison

$$\begin{aligned}
\hat{x}_{US}(k+1\,|k) &= A(k)\hat{x}_{US}(k\,|k-1) + L_{US}(k)\left(y(k) - C(k)\hat{x}_{US}(k\,|k-1)\right) \\
L_{US}(k) &= \left(A(k)P_{US}(k)C^T(k) + E(k)F^T(k)\right)R_{US}^{-1}(k), \\
P_{US}(k+1) &= A(k)P_{US}(k)A^T(k) - L_{US}(k)R_{US}(k)L_{SU}^T(k) \\
&\quad + E(k)E^T(k), \\
R_{US}(k) &= F(k)F^T(k) + C(k)P_{US}(k)C^T(k), \\
\hat{x}_{LS}(k+1\,|k) &= A(k)\hat{x}_{LS}(k\,|k-1) + L_{LS}(k)\left(y(k) - C(k)\hat{x}_{LS}(k\,|k-1)\right), \\
L_{LS}(k) &= A(k)P_{LS}(k)C^T(k)R_{LS}^{-1}(k), \\
P_{LS}(k+1) &= A(k)P_{LS}(k)A^T(k) - L_{LS}(k)R_{LS}(k)L_{LS}^T(k) + W_2^{-1}(k), \\
R_{LS}(k) &= F(k)F^T(k) + C(k)P_{LS}(k)C^T(k);
\end{aligned}$$

- fault estimation in comparison

$$\begin{aligned}
\hat{f}_{US}(k) &= F^T(k)R_{US}^{-1}(k)\left(y(k) - \hat{x}_{US}(k\,|k-1)\right), \\
\hat{f}_{LS}(k) &= F^T(k)R_{LS}^{-1}(k)\left(y(k) - \hat{x}_{LS}(k\,|k-1)\right).
\end{aligned}$$

For the comparison sake, it is supposed that

$$E(k) = I, W_2(k) = I.$$

It is clear that both estimation schemes have the similar form for the one-step prediction of $x(k)$ and the fault vector. The difference lies in the observer gain matrix, and associated with it, in the solution of the Riccati recursion. Note that both $P_{US}(k)$ and $P_{LS}(k)$ have the similar interpretation associated with the estimation performance, but their updates in the Riccati recursions are slightly different. In fact, the difference in the observer gain matrix can also be interpreted whether the estimation of the fault vector $f(k)$ is included in the update of the state estimation from k to $k+1$.

For our fault diagnosis study, a reasonable comparison basis for both estimation schemes is the properties of the residual vectors. Our previous work reveals that

$$\|r_{LS}(k)\|_{2,[0,N]}^2 \le e_x^T(0)P^{-1}(0)e_x(0) + \|f(k)\|_{2,W_f(k),[0,N]}^2, \tag{8.58}$$

$$\|r_{US}(k)\|_{2,[0,N]}^2 \le e_x^T(0)P^{-1}(0)e_x(0) + \|f(k)\|_{2,[0,N]}^2, \tag{8.59}$$

$$r_{LS}(k) = R_{LS}^{-1/2}(k)\left(y(k) - C(k)\hat{x}_{LS}(k\,|k-1)\right),$$

$$r_{US}(k) = R_{US}^{-1/2}(k)\left(y(k) - C(k)\hat{x}_{US}(k\,|k-1)\right).$$

On the assumption that the term $e_x^T(0)P^{-1}(0)e_x(0)$ becomes sufficiently small in comparison with $\|f(k)\|_{2,[0,N]}^2$ for a large N, it becomes evident from (8.58)–(8.59) that $\|r_{US}(k)\|_{2,[0,N]}^2$ gives a lower bound of the l_2-norm of the fault vector, while the relation between $\|r_{LS}(k)\|_{2,[0,N]}^2$ and the l_2 -norm of the fault vector depends on the weighting matrices $W_1(k)$ and $W_2(k)$. When $W_1(k)$ and $W_2(k)$ are selected according to (8.48)–(8.49), (8.58) becomes

$$\|r_{LS}(k)\|_{2,[0,N]}^2 \le e_x^T(0)P^{-1}(0)e_x(0) + \|f(k)\|_{2,[0,N]}^2.$$

Moreover, it is of interest to notice that it holds, in both cases,

$$\left\|\hat{f}_{LS}(k)\right\|_{2,[0,N]}^2 \le \|r_{LS}(k)\|_{2,[0,N]}^2,$$

$$\left\|\hat{f}_{US}(k)\right\|_{2,[0,N]}^2 \le \|r_{US}(k)\|_{2,[0,N]}^2.$$

In other words, in both cases, the l_2-norm of the residual vector gives a better estimation of the l_2-norm of fault vector $f(k)$.

8.4 Least Squares Observer for Process Fault Estimation

We now consider fault estimation issues on the assumption of the LDTV system model

$$\begin{aligned} x(k+1) &= A(k)x(k) + E(k)f(k), \\ y(k) &= C(k)x(k) + v(k), \end{aligned}$$

where $x(k) \in \mathcal{R}^n, y(k) \in \mathcal{R}^m, v(k) \in \mathcal{R}^m$ are process state, output and disturbance vectors, respectively. $f(k) \in \mathcal{R}^{k_f}$ is the unknown fault vector to be estimated, which represents process faults. We formulate the fault estimation problem as finding $x(k), k = 0, \cdots, N, f(k), k = 0, \cdots, N-1$, such that for given measurement data $y(0), \cdots, y(k), \cdots, y(N)$ the cost function

$$J_N = \frac{1}{2}\begin{pmatrix} \|x(0) - x_o\|^2_{P^{-1}(0)} + \|y(k) - C(k)x(k)\|^2_{2,[0,N]} + \\ \|x(k+1) - A(k)x(k)\|^2_{2,[0,N-1]} \end{pmatrix} \tag{8.60}$$

$$\text{s.t. } x(k+1) = A(k)x(k) + E(k)f(k), \tag{8.61}$$

$$y(k) = C(k)x(k) + v(k) \tag{8.62}$$

is minimised, where

$$rank\,(E(k)) = k_f,$$

$P(0) > 0$ is given and x_o is an initial estimation for $x(0)$ based on *a priori* knowledge.

Remark 8.3 *Considering that the fault vector is only present in the system state equation, we call it process fault.*

We now summarise the problem solution in the following theorem.

Theorem 8.3 *The solution of the optimisation problem (8.60) is given by*

$$\hat{x}(k\,|k) = \hat{x}(k\,|k-1) + L(k\,|k)\left(y(k) - C(k)\hat{x}(k\,|k-1)\right), \tag{8.63}$$

$$\hat{x}(k+1\,|k) = A(k)\hat{x}(k\,|k), \hat{x}(0\,|-1) := x_o, \tag{8.64}$$

$$\hat{f}(k) = L_f(k)\left(y(k+1) - C(k+1)\hat{x}(k+1\,|k)\right), \tag{8.65}$$

$$L(k\,|k) = P(k\,|k-1)C^T(k)R^{-1}(k), \tag{8.66}$$

$$R(k) = I + C(k)P(k\,|k-1)C^T(k), \tag{8.67}$$

$$L_f(k) = \left(E(k)E^T(k)\right)^{-1} E^T(k)C^T(k+1)R^{-1}(k+1), \tag{8.68}$$

$$P(k+1\,|k) = I + A(k)P(k\,|k)A^T(k), \tag{8.69}$$

$$P^{-1}(k+1\,|k+1) = P^{-1}(k+1\,|k) + C^T(k+1)C(k+1). \tag{8.70}$$

Moreover,

$$J_N = J(N) + \left\| x(N) - \hat{x}(N\,|N\,) \right\|^2_{P^{-1}(N|N\,)}, \tag{8.71}$$

$$\min_{x(k), f(k), k=0,\cdots N-1, x(N)} J_N = J(N) + \min_{x(N)} \left\| x(N) - \hat{x}(N\,|N\,) \right\|^2_{P^{-1}(N|N\,)} \tag{8.72}$$

$$= J(N) = \frac{1}{2} \left\| y(k) - C(k)\hat{x}(k\,|k-1\,) \right\|^2_{R^{-1}(k),2,[0,N]}. \tag{8.73}$$

Proof The theorem will be proved using the induction method. To this end, consider, at first, $N = 1$, that is

$$\min_{x(0), x(1), f(0)} J_1,$$

$$J_1 = \frac{1}{2} \begin{pmatrix} \|x(0) - x_o\|^2_{P^{-1}(0)} + \|y(0) - C(0)x(0)\|^2 + \|E(0)f(0)\|^2 \\ + \|y(1) - C(1)x(1)\|^2 \end{pmatrix},$$

$$\text{s.t. } x(k+1) = A(k)x(k) + E(k)f(k)$$

$$y(k) = C(k)x(k) + v(k).$$

Along the lines of the study in Sect. 8.2, we have

$$J_1 = J(0) + \frac{1}{2} \begin{pmatrix} \left\| x(0) - \hat{x}(0\,|0\,) \right\|^2_{P^{-1}(0|0\,)} + \|E(0)f(0)\|^2 \\ + \|y(1) - C(1)x(1)\|^2 \end{pmatrix},$$

$$J(0) = \frac{1}{2} \left\| \hat{x}(0\,|0\,) - x_o \right\|^2_{P^{-1}(0)} + \frac{1}{2} \left\| y(0) - C(0)\hat{x}(0\,|0\,) \right\|^2$$

$$= \frac{1}{2} \left\| y(0) - C(0)\hat{x}(0\,|-1\,) \right\|^2_{R^{-1}(0)}, \hat{x}(0\,|-1\,) = x_o,$$

$$\hat{x}(0\,|0\,) = x_o + L(0\,|0\,) \left(y(0) - C(0)x_o \right),$$

$$L(0\,|0\,) = P(0)C^T(0)R^{-1}(0), R(0) = I + C(0)P(0)C^T(0),$$

$$P^{-1}(0\,|0\,) = P^{-1}(0) + C^T(0)C(0).$$

Viewing

$$\min_{x(0), f(0)} \frac{1}{2} \left(\left\| x(0) - \hat{x}(0\,|0\,) \right\|^2_{P^{-1}(0|0\,)} + \|E(0)f(0)\|^2 \right)$$

$$\text{s.t. } x(1) = A(0)x(0) + E(0)f(0)$$

as a RLS estimation problem yields

$$
\begin{gathered}
x(0)-\hat{x}(0\,|0\,)=P(0\,|0\,)A^T(0)Q^{-1}\left(x(1)-A(0)\hat{x}(0\,|0\,)\right),\\
Q=I+A(0)P(0\,|0\,)A^T(0),\\
E(0)f(0)=x(1)-A(0)x(0)=Q^{-1}\left(x(1)-A(0)\hat{x}(0\,|0\,)\right)\Longrightarrow\\
\min_{x(0),f(0)}\frac{1}{2}\left(\left\|x(0)-\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(0|0\,)}+\|E(0)f(0)\|^2\right)\\
=\frac{1}{2}\left\|x(1)-A(0)\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(1|0\,)},\\
P(1\,|0\,):=Q=I+A(0)P(0\,|0\,)A^T(0).
\end{gathered}
$$

Thus, it turns out

$$
\begin{gathered}
\min_{x(1)}J_1=J(0)+\min_{x(1)}\frac{1}{2}\left(\left\|x(1)-A(0)\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(1|0\,)}+\|y(1)-C(1)x(1)\|^2\right)\\
\text{s.t. } y(1)=C(1)x(1)+v(1),
\end{gathered}
$$

which is solved by

$$
\begin{aligned}
&\hat{x}(1\,|1\,)=\hat{x}(1\,|0\,)+L(1\,|1\,)\left(y(1)-C(1)\hat{x}(1\,|0\,)\right),\\
&L(1\,|1\,)=P(1\,|0)C^T(1)R^{-1}(1),\\
&\hat{x}(1\,|0\,)=A(0)\hat{x}(0\,|0\,),\ R(1)=I+C(0)P(1\,|0)C^T(0),
\end{aligned}
$$

and results in

$$
\begin{aligned}
\min_{x(1)}J_1&=J(1)+\min_{x(1)}\left\|x(1)-\hat{x}(1\,|1\,)\right\|^2_{P^{-1}(1|1\,)}=J(1),\\
J(1)&=J(0)+\frac{1}{2}\left(\begin{array}{c}\left\|\hat{x}(1\,|1\,)-A(0)\hat{x}(0\,|0\,)\right\|^2_{P^{-1}(1|0\,)}+\\ \left\|y(1)-C(1)\hat{x}(1\,|1\,)\right\|^2\end{array}\right)\\
&=\frac{1}{2}\left(\left\|y(0)-C(0)\hat{x}(0\,|-1\,)\right\|^2_{R^{-1}(0)}+\left\|y(1)-C(1)\hat{x}(1\,|0\,)\right\|^2_{R^{-1}(1)}\right).
\end{aligned}
$$

Note that the optimal estimation for $f(0)$ is given by

$$
\begin{gathered}
E(0)f(0)=P^{-1}(1\,|0\,)\left(x(1)-A(0)\hat{x}(0\,|0\,)\right)\Longrightarrow\\
E(0)\hat{f}(0)=C^T(1)R^{-1}(1)\left(y(1)-C(1)\hat{x}(1\,|0)\right)\Longrightarrow\\
\hat{f}(0)=\left(E^T(0)E(0)\right)^{-1}E^T(0)C^T(1)R^{-1}(1)\left(y(1)-C(1)\hat{x}(1\,|0)\right).
\end{gathered}
$$

It is obvious that for $N=1$ the results given in (8.63)–(8.73) are proved. Now, we check the case for $N=k+1$ on the assumption that (8.63)–(8.73) hold for $N=k$. We begin with

$$
J_{k+1}=J_k+\|E(k)f(k)\|^2+\|y(k+1)-C(k+1)x(k+1)\|^2,
$$

which can be further written as

$$\begin{gathered}
J_{k+1} = J(k) + \left\| x(k) - \hat{x}(k\,|k\,) \right\|^2_{P^{-1}(k|k\,)} \\
+ \left\| E(k)f(k) \right\|^2 + \left\| y(k+1) - C(k+1)x(k+1) \right\|^2, \\
J(k) = \frac{1}{2} \left\| y(i) - C(i)\hat{x}(i\,|i-1\,) \right\|^2_{R^{-1}(i),2,[0,k]}, \\
\text{s.t. } x(k+1) = A(k)x(k) + E(k)f(k), \\
y(k+1) = C(k+1)x(k+1) + v(k+1).
\end{gathered}$$

Analogue to the study on case $N = 1$, solving the RLS problem for

$$\begin{gathered}
\left\| x(k) - \hat{x}(k\,|k\,) \right\|^2_{P^{-1}(k|k\,)} + \left\| E(k)f(k) \right\|^2 \\
\text{s.t. } x(k+1) = A(k)x(k) + E(k)f(k)
\end{gathered}$$

leads to

$$\begin{gathered}
\min_{x(i),f(i),i=1,\cdots,k,x(k+1)} J_{k+1} = J(k) + \\
\min_{x(k+1)} \left\| x(k+1) - A(k)\hat{x}(k\,|k\,) \right\|^2_{P^{-1}(k+1|k\,)} + \left\| y(k+1) - C(k+1)x(k+1) \right\|^2 \\
\text{s.t. } y(k+1) = C(k+1)x(k+1) + v(k+1),
\end{gathered}$$

which is finally solved with the result

$$\begin{gathered}
J_{k+1} = J(k+1) + \left\| x(k+1) - \hat{x}(k+1\,|k+1\,) \right\|^2_{P^{-1}(k+1|k+1\,)} \Longrightarrow \\
\min_{x(i),f(i),i=1,\cdots,k,x(k+1)} J_{k+1} = J(k+1) \\
= \frac{1}{2} \left\| y(i) - C(i)\hat{x}(i\,|i-1\,) \right\|^2_{R^{-1}(i),2,[0,k+1]}, \\
\hat{x}(k+1\,|k+1\,) = \hat{x}(k+1\,|k\,) + L(k+1\,|k+1\,) \begin{pmatrix} y(k+1)- \\ C(k+1)\hat{x}(k+1\,|k\,) \end{pmatrix}, \\
\hat{x}(k+1\,|k\,) = A(k)\hat{x}(k\,|k\,), \\
L(k+1\,|k+1\,) = P(k+1\,|k\,)C^T(k+1)R^{-1}(k+1), \\
P(k+1\,|k\,) = I + A(k)P(k\,|k\,)A^T(k), \\
R(k+1) = I + C(k+1)P(k+1\,|k\,)C^T(k+1),
\end{gathered}$$

as well as

$$
\begin{aligned}
E(k)\hat{f}(k) &= C^T(k+1)R^{-1}(k+1)\left(y(k+1) - C(k+1)\hat{x}(k+1\,|k\,)\right) \\
\Longrightarrow \hat{f}(k) &= L_f(k)\left(y(k+1) - C(k+1)\hat{x}(k+1\,|k\,)\right), \\
L_f(k) &= \left(E^T(k)E(k)\right)^{-1} E^T(k)C^T(k+1)R^{-1}(k+1).
\end{aligned}
$$

Thus, (8.63)–(8.73) hold for $N = k + 1$. The theorem is proved.

It is straightforward that the update of $P(k\,|k-1\,)$, $P(k\,|k\,)$ can also be done using the following Riccati recursions:

$$
\begin{gathered}
P(k+1\,|k\,) = I + A(k)P(k\,|k-1\,)A^T(k) - L(k+1\,|k\,)R(k)L^T(k+1\,|k\,), \\
L(k+1\,|k\,) = A(k)L(k\,|k\,), \\
P(k+1\,|k+1\,) = I + A(k)P(k\,|k\,)A^T(k) - \Psi(k), \\
\Psi(k) = L(k+1\,|k+1\,)R(k+1)L^T(k+1\,|k+1\,).
\end{gathered}
$$

It is evident that it holds

$$
\begin{gathered}
\left\| y(k) - C(k)\hat{x}(k\,|k-1\,) \right\|^2_{R^{-1}(k),2,[0,N]} \le \\
\|x(0) - x_o\|^2_{P^{-1}(0)} + \|v(k)\|^2_{2,[0,N]} + \|E(k)f(k)\|^2_{2,[0,N-1]}\,.
\end{gathered}
$$

When $\|x(0) - x_o\|^2_{P^{-1}(0)} + \|v(k)\|^2_{2,[0,N]}$ is sufficiently small, we also have

$$
\left\| y(k) - C(k)\hat{x}(k\,|k-1\,) \right\|^2_{R^{-1}(k),2,[0,N]} \le \|E(k)f(k)\|^2_{2,[0,N-1]}\,.
$$

8.5 Notes and References

Fault estimation in dynamic systems is receiving considerable attention in the research field of fault diagnosis and fault-tolerant control. This trend is highly motivated by the argument that the estimate of a fault can be directly applied for fault detection and further, when dealing with fault-tolerant control, used for achieving fault compensation as well. As demonstrated in the previous chapters, the use of fault estimate for fault detection may lead to poor performance, in particular, when uncertainties exist in the process or process model under consideration. In the last part of this book, we will also investigate fault-tolerant control issues and give a critical review of fault compensation based fault-tolerant control strategies.

Observer-based fault estimation schemes are the state of the art in research. One popular design strategy is the application of the robust unknown input observer (UIO) technique. The robustness is in general achieved in the l_2-gain optimisation framework, which can be roughly formulated as

$$
\left\| f - \hat{f} \right\|_2 \le \gamma \|d\|_2
$$

with d representing the unknown input vector. In this context, a fault estimator is designed to minimise the l_2-gain γ with respect to d. It is beyond dispute that in the l_2-gain optimisation framework many well-established mathematical and control theoretical methods can be used as a tool to deal with fault estimation issues for various types of systems. The reader is referred to the first publications in this thematic field [1–5], which are helpful to understand the basic ideas and the applied tools. Moreover, it can be observed that this technique has also been adopted in the integrated design of robust controller and FD systems, as proposed in [6–8]. On the other hand, it is worth mentioning that this fault estimation strategy and its performance have been critically reviewed in [9].

In the past two decades, application of augmented observer schemes to fault estimation has received increasing attention. The underlying idea of the augmented observer schemes lies in addressing the faults to be estimated as additional state variables, which are then re-constructed by an augmented observer. The well-known PI-observer is a special kind of such observers [10, 11]. The augmented observer technique is strongly related to the UIO scheme. In this context, the augmented observer is also called *simultaneous state and disturbance estimator* [12]. Often, such observers/estimators are designed based on certain assumption on the faults, for instance the boundedness on the derivative. We refer the reader to [13–17] for some representative publications on this topic.

In this chapter, we have investigated fault estimation issues from the "least squares" optimisation viewpoint, which is considerably different from the robust unknown input observer and the augmented observer schemes. The so-called least squares observers are in fact the analogue form of the celebrated Kalman filter and can be applied to the estimation of state variables in processes with deterministic unknown inputs. Intensive studies on this topic have been reported in the literature in 1970's, as Kalman filter theory was successfully established [18, 19] . Unfortunately, in recent research, few attention has been paid to such type of optimal observers and their potential applications, for instance, in fault detection area. Our work on this topic has been remarkably motivated by the unified solution for optimal fault detection and the associated results presented in the last chapter. A Willems's paper on "deterministic least squares filtering" [20] has inspired the formulation of fault estimation as a least squares optimisation problem and its interpretation in the context of fault estimation.

The mathematical tool for the solution of our least squares estimation is the regularised least squares estimation method, in which *a priori* knowledge of the variables to be estimated is embedded in the optimisation. This handling allows us to extend the static RLS estimation to dynamic processes. Moreover, the minimisation of the term $\|y - Hx\|_W^2$ (see equation (8.6)) results in an estimation for

$$v = y - Hx,$$

which is of the minimum norm (least squares). That is,

$$\left\|\hat{v}\right\|_2 \leq \|v\|_2 ,$$

which can be used for estimating the the lower bound of the l_2-norm of the fault vector. Concerning the RLS method, we refer the reader to [21] for a systematic description.

To our knowledge, there are (very) few publications on LS estimation for LDTV systems with deterministic unknown inputs formulated in (8.16)–(8.19). Under this consideration, we have described the solution and, above all, the procedure of the solution in details. We have studied two LS fault estimation problems, one for estimating sensor type of faults and the other for process faults. Moreover, we have briefly discussed about the relations between the unified solution (with faults as unknown inputs) and the LS estimation algorithms given in this chapter.

The formulation of fault estimation as a RLS optimisation problem builds the basis for applying the existing optimisation techniques to addressing issues like online optimisation of fault detection systems. In that case, an alternative solution form will be adopted, as introduced in Chap. 20. On the other hand, dealing with uncertainties remains an open and challenging issue.

Finally, we would like to mention that model-based fault estimation is a vital research area. The so-called parameter identification technique (PIT) based fault estimation (identification) builds, in parallel with the observer-based strategy, one of the mainstreams in this research area. The core of PIT-based fault estimation consists in the application of the well-established parameter identification technique to the identification of the faults that are modelled as system parameters. This technique is especially efficient in dealing with multiplicative faults. We refer the interested reader to [22–26] for a comprehensive study of this technique. Further active fields in the thematic area of fault estimation include, for example, sliding mode observer-based fault detection and estimation [27–29], strong tracking filter technique for fault and parameter estimation [30].

References

1. H. Niemann, A. Saberi, A. Stoovogel, and P. Sannuti, "Optimal fault estimation," *Proc. of the 4th IFAC Symp. SAFEPROCESS*, vol. 1, pp. 262–267, 2000.
2. H. Niemann and J. J. Stoustrup, "Design of fault detectors using h-infinity optimization," *Proc. of the 39th IEEE CDC*, 2000.
3. A. Saberi, A. Stoovogel, P. Sannuti, and H. Niemann, "Fundamental problems in fault detection and identification," *Int. J. Robust Nonlinear Contr.*, vol. 10, pp. 1209–1236, 2000.
4. H. Wang and G.-H. Yang, "Fault estimations for uncertain linear discrete-time systems in low frequency domain," *Proc. of the 2007 ACC*, pp. 1124–1129, 2007.
5. H. Wang and G.-H. Yang, "Fault estimations for linear systems with polytopic uncertainties," *Int. J. Systems, Control and Communications*, vol. 1, pp. 53–71, 2008.
6. G. Murad, I. Postlethwaite, and D.-W. Gu, "A robust design approach to integrated control and diagnostics," *Proc. of the 13th IFAC Word Congress*, vol. 7, pp. 199–204, 1996.
7. C. N. Nett, C. Jacobson, and A. T. Miller, "An integrated approach to controls and diagnostics," *Proc. of ACC*, pp. 824–835, 1988.
8. M. L. Tyler and M. Morari, "Optimal and robust design of integrated control and diagnostic modules," *Proc. of ACC*, pp. 2060–2064, 1994.

9. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms, and Tools*. Springer-Verlag, 2008.
10. K. Busawon and P. Kabore, “Disturbance attenuation using proportional integral observers,” *Int. J. Contr.*, vol. 74, pp. 618–627, 2001.
11. M. Saif, “Reduced-order proportional integral observer with application,” *J. Guidance control dynamics*, vol. 16, pp. 985–988, 1993.
12. B. Shafai, C. T. Pi, and S. Nork, “Simultaneous disturbance attenuation and fault detection using proportional integral observers,” *Proc. ACC*, pp. 1647–1649, 2002.
13. Z. Gao and D. Ho, “Proportional multiple-integral observer design for descriptor systems with measurement output disturbances,” *IEE Proc. - Control Theory Appl.*, vol. 151(3), pp. 279–288, 2004.
14. Z. Gao and D. W. C. Ho, “State/Noise estimator for descriptor systems with application to sensor fault diagnosis,” *IEEE Trans. Signal Processing*, vol. 54, pp. 1316–1326, 2006.
15. Q. P. Ha and H. Trinh, “State and input simultaneous estimation for a class of nonlinear systems,” *Automatica*, vol. 40, pp. 1779–1785, 2004.
16. Z. Gao and S. X. Ding, “Actuator fault robust estimation and fault-tolerant control for a class of nonlinear descriptor systems,” *Automatica*, vol. 43, pp. 912–920, 2007.
17. Z. Gao, X. Shi, and S. Ding, “Fuzzy state/disturbance observer design for T-S fuzzy systems with application to sensor fault estimation,” *IEEE Trans. on Syst. Man and Cyber - Part B, Cybernetics*, vol. 38, pp. 875–880, 2008.
18. H. W. Sorenson, “Least-squares estimation: From Gauss to Kalman,” *IEEE Spectrum*, pp. 63–68, 1970.
19. P. Swerling, “Modern state estimation methods from the viewpoint of the method of least squares,” *IEEE Trans. on Autom. Contr.*, vol. 16, pp. 707–719, 1971.
20. J. Willems, “Deterministic least squares filtering,” *Journal of Econometrics*, vol. 118, pp. 341–373, 2004.
21. T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. New Jersey: Prentice Hall, 1999.
22. J. Gertler, “Survey of model-based failure detection and isolation in complex plants,” *IEEE Control Systems Magazine*, vol. 3, pp. 3–11, 1988.
23. R. Isermann, “Process fault detection based on modeling and estimation methods - a survey,” *Automatica*, vol. 20, pp. 387–404, 1984.
24. R. Isermann, “Supervision, fault-detection and fault-diagnosis methods -an introduction,” *Control Engineering Practice*, vol. 5 (5), pp. 639–652, 1997.
25. R. Isermann, *Fault Diagnosis Systems*. Berlin Heidelberg: Springer-Verlag, 2006.
26. S. Simani, S. Fantuzzi, and R. J. Patton, *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. London: Springer-Verlag, 2003.
27. C. Tan and C. Edwards, “Sliding mode observers for detection and reconstruction of sensor faults,” *Automatica*, vol. 38, pp. 1815–1821, 2002.
28. W. Chen and M. Saif, “A sliding mode observer-based strategy for fault detection, isolation, and estimation in a class of lipschitz nonlinear systems,” *Int. J. of Syst. Science*, vol. 38, pp. 943–955, 2007.
29. H. Alwi, C. Edwards, and C. P. Tan, *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Springer-Verlag, 2011.
30. D. H. Zhou and P. M. Frank, “Strong tracking filtering of nonlinear time-varying stochastic systems with coloured noise: Application to parameter estimation and empirical robustness analysis,” *Int. J. Control*, vol. 65, pp. 295–307, 1996.

Chapter 9
Detection and Isolation of Multiplicative Faults

The previous chapters are mainly dedicated to the diagnosis issues of additive faults. Even if further efforts are needed to develop novel methods to deal with this class of faults more efficiently, it is the common opinion that the framework of diagnosing additive faults is well established. Differently, detecting and isolating multiplicative faults are challenging and open issues that are of significant research and practical interests. Multiplicative faults, also those with small size, may cause remarkable changes in the system structure and dynamics. Often, they rise up in a continuing process, which hinders, different from those rapid changes, an early and reliable detection, in particular, when these faults are embedded in a closed-loop control system. In this chapter, we focus on issues of detecting multiplicative faults in the open- and closed-loop system configurations. At the end of this chapter, we also deal with isolation of multiplicative faults.

9.1 System Modelling

9.1.1 Model Forms

We consider LTI systems of the form

$$y(s) = G(s)u(s),\, y \in \mathcal{C}^m,\, u \in \mathcal{C}^p \tag{9.1}$$

with minimal state space realisation

$$G(s) = (A, B, C, D)\,, \tag{9.2}$$

where A, B, C, D are system matrices of appropriate dimensions. The LCF and RCF of $G(s)$ are given by

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_9

$$G(s) = \hat{M}^{-1}(s)\hat{N}(s) = N(s)M^{-1}(s), \tag{9.3}$$

respectively, where $\hat{M}(s) \in \mathcal{RH}_\infty^{m\times m}$, $\hat{N}(s) \in \mathcal{RH}_\infty^{m\times p}$, $M(s) \in \mathcal{RH}_\infty^{p\times p}$, $N(s) \in \mathcal{RH}_\infty^{m\times p}$. $\left(\hat{M}(s), \hat{N}(s)\right)$ and $(M(s), N(s))$ are left and right coprime pairs over $\mathcal{RH}_\infty$, for which there exist $\hat{X}(s) \in \mathcal{RH}_\infty^{m\times m}$, $\hat{Y}(s) \in \mathcal{RH}_\infty^{p\times m}$, $X(s) \in \mathcal{RH}_\infty^{p\times p}$, $Y(s) \in \mathcal{RH}_\infty^{p\times m}$ so that

$$\begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} \begin{bmatrix} -\hat{Y} \\ \hat{X} \end{bmatrix} = I_{m\times m}, \begin{bmatrix} X & Y \end{bmatrix} \begin{bmatrix} M \\ N \end{bmatrix} = I_{p\times p}. \tag{9.4}$$

With the aid of the above coprime factorisations of LTI systems, we are now in a position to introduce the system models adopted in our work. We denote the nominal (fault- and uncertainty-free) plant model, the faulty model and plant model with uncertainty as well as their LCF and RCF by

$$G_o(s) = \hat{M}_o^{-1}(s)\hat{N}_o(s) = N_o(s)M_o^{-1}(s), \tag{9.5}$$

$$G_f(s) = \hat{M}_f^{-1}(s)\hat{N}_f(s) = N_f(s)M_f^{-1}(s), \tag{9.6}$$

$$G_\Delta(s) = \hat{M}_\Delta^{-1}(s)\hat{N}_\Delta(s) = N_\Delta(s)M_\Delta^{-1}(s), \tag{9.7}$$

respectively. The LC and RC pairs $(\hat{M}_o, \hat{N}_o)$ and (M_o, N_o) are called normalised, if

$$\begin{bmatrix} \hat{M}_o(s) & \hat{N}_o(s) \end{bmatrix} \begin{bmatrix} \hat{M}_o(s) & \hat{N}_o(s) \end{bmatrix}^* = I, \begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix}^* \begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix} = I.$$

For the state space computation of the normalised LC and RC pairs, the following theorem is well-known. The reader is referred to the references given at the end of this chapter.

Theorem 9.1 *Given the system model (9.1) with minimal state space realisation (9.2), then*

$$\begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix} = \begin{bmatrix} \Gamma^{-1/2} \\ D\Gamma^{-1/2} \end{bmatrix} + \begin{bmatrix} F_o \\ C_{F_o} \end{bmatrix} \left(sI - A_{F_o}\right)^{-1} B\Gamma^{-1/2},$$

$$\begin{bmatrix} \hat{M}_o(s) & \hat{N}_o(s) \end{bmatrix} = \begin{bmatrix} \bar{\Gamma}^{-1/2} & \bar{\Gamma}^{-1/2}D \end{bmatrix} + \bar{\Gamma}^{-1/2}C\left(sI - A_{L_o}\right)^{-1} \begin{bmatrix} -L_o & B_{L_o} \end{bmatrix}$$

build the normalised RC and LC pair of G, respectively, where

$$A_{F_o} = A + BF_o, A_{L_o} = A - L_oC, B_{L_o} = B - L_oC, C_{F_o} = C + DF_o,$$
$$\Gamma = I + D^TD, \bar{\Gamma} = I + DD^T,$$
$$F_o = -\Gamma^{-1}\left(B^TX + D^TC\right), L_o = \left(BD^T + YC^T\right)\bar{\Gamma}^{-1}$$

with $X \geq 0$, $Y \geq 0$ being the solutions of the following Riccati equations

$$A_X^T X + X A_X - X B \Gamma^{-1} B^T X + C^T \bar{\Gamma}^{-1} C = 0, A_X = A - B \Gamma^{-1} D^T C,$$
$$A_Y Y + Y A_Y^T - X C^T \bar{\Gamma}^{-1} C X + B \Gamma^{-1} B^T = 0, A_Y = A - B D^T \bar{\Gamma}^{-1} C.$$

For practical applications, the LCF and RCF of $G_f(s)$, $G_\Delta(s)$ are often expressed in the form given below:

$$\begin{aligned}
\hat{M}_f(s) &= \hat{M}_o(s) + \Delta_{\hat{M}_f}(s), \hat{N}_f(s) = \hat{N}_o(s) + \Delta_{\hat{N}_f}(s),\\
M_f(s) &= M_o(s) + \Delta_{M_f}(s), N_f(s) = N_o(s) + \Delta_{N_f}(s),\\
\hat{M}_\Delta(s) &= \hat{M}_o(s) + \Delta_{\hat{M}}(s), \hat{N}_\Delta(s) = \hat{N}_o(s) + \Delta_{\hat{N}}(s),\\
M_\Delta(s) &= M_o(s) + \Delta_M(s), N_\Delta(s) = N_o(s) + \Delta_N(s),
\end{aligned}$$

where

$$\Delta_{\hat{M}_f}(s), \Delta_{\hat{N}_f}(s), \Delta_{M_f}(s), \Delta_{N_f}(s), \Delta_{\hat{M}}(s), \Delta_{\hat{N}}(s), \Delta_M(s), \Delta_N(s) \in \mathcal{RH}_\infty$$

are some unknown transfer functions. $\Delta_{\hat{M}_f}, \Delta_{\hat{N}_f}, \Delta_{M_f}, \Delta_{N_f}$ represent multiplicative faults, when they are different from zero. There are numerous ways to model these terms in more details, depending on available *a priori* knowledge. For instance,

$$\Delta_{\hat{M}_f}(s) = \hat{M}_o(s)\delta_{\hat{M}_f}(s), \Delta_{\hat{N}_f}(s) = \hat{N}_o(s)\delta_{\hat{N}_f}(s), \delta_{\hat{M}_f}(s), \delta_{\hat{N}_f}(s) \in \mathcal{RH}_\infty$$

represent those faults with known and unknown parts, denoted by $\left(\hat{M}_o, \hat{N}_o\right)$ and $\left(\delta_{\hat{M}_f}, \delta_{\hat{N}_f}\right)$, respectively. It is also often the case that we only know the boundedness of uncertainties, for instance, expressed by

$$\left\| \begin{matrix} \Delta_{\hat{N}}(s) & \Delta_{\hat{M}}(s) \end{matrix} \right\|_\infty \leq \delta_\Delta.$$

Remember that SKR and SIR are alternative system representations, which are related to the LCF and RCF of a system under consideration. Let

$$\mathcal{K}_o = \begin{bmatrix} -\hat{N}_o(s) & \hat{M}_o(s) \end{bmatrix}, \mathcal{I}_o = \begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix}$$

be the SKR and SIR of the nominal system, which are possibly normalised, and

$$\begin{aligned}
\mathcal{K}_f &= \begin{bmatrix} -\hat{N}_f(s) \ \hat{M}_f(s) \end{bmatrix}, \mathcal{I}_f = \begin{bmatrix} M_f(s) \\ N_f(s) \end{bmatrix},\\
\mathcal{K}_\Delta &= \begin{bmatrix} -\hat{N}_\Delta(s) \ \hat{M}_\Delta(s) \end{bmatrix}, \mathcal{I}_\Delta = \begin{bmatrix} M_\Delta(s) \\ N_\Delta(s) \end{bmatrix}
\end{aligned}$$

denote the SKR and SIR of the faulty and uncertain system models, respectively. In our subsequent work, we may also suppose

$$\mathcal{K}_f = \begin{bmatrix} -\hat{N}_o(s) & \hat{M}_o(s) \end{bmatrix} \left(I + \Delta_{\mathcal{K}_f}\right), \mathcal{I}_f = \left(I + \Delta_{\mathcal{I}_f}\right) \begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix}, \tag{9.8}$$

$$\mathcal{K}_\Delta = \begin{bmatrix} -\hat{N}_o(s) & \hat{M}_o(s) \end{bmatrix} (I + \Delta_{\mathcal{K}}), \mathcal{I}_\Delta = (I + \Delta_{\mathcal{I}}) \begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix}, \tag{9.9}$$

where $\Delta_{\mathcal{K}_f}, \Delta_{\mathcal{I}_f}, \Delta_{\mathcal{K}}, \Delta_{\mathcal{I}} \in \mathcal{RH}_\infty$ are unknown and assumed to be bounded, for instance

$$\left\|\Delta_{\mathcal{K}_f}\right\|_\infty < 1, \left\|\Delta_{\mathcal{I}_f}\right\|_\infty < 1, \|\Delta_{\mathcal{K}}\|_\infty < 1, \|\Delta_{\mathcal{I}}\|_\infty < 1.$$

9.1.2 Relations Among the Model Forms

We now investigate the relations among the model forms presented in the previous sub-section. Considering that the model uncertainties and multiplicative faults are handled in an analogue manner in the models introduced above, we will first address them uniformly as uncertainties. The following lemmas provide us with the equivalence between different model forms.

Lemma 9.1 *Consider an LTI system with the nominal model $G_o(s)$ and its extended form $G(s)$, including uncertainties or faults. Let*

$$G_o(s) = \hat{M}_o^{-1}(s)\hat{N}_o(s) = N_o(s)M_o^{-1}(s),$$
$$G(s) = \hat{M}^{-1}(s)\hat{N}(s) = N(s)M^{-1}(s)$$

be their LCFs and RCFs. Then, the SKR and SIR of $G(s)$,

$$\mathcal{K} = \begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix}, \mathcal{I} = \begin{bmatrix} M \\ N \end{bmatrix},$$

can be equivalently written as

$$\begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} = \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} (I + \Delta_{\mathcal{K}}), \begin{bmatrix} M \\ N \end{bmatrix} = (I + \Delta_{\mathcal{I}}) \begin{bmatrix} M_o \\ N_o \end{bmatrix},$$

for some $\Delta_{\mathcal{K}}, \Delta_{\mathcal{I}} \in \mathcal{RH}_\infty$.

Proof Let

$$\Delta_{\mathcal{K}} = \begin{bmatrix} -\hat{Y}_o \\ \hat{X}_o \end{bmatrix} \left(\begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} - \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} \right) \in \mathcal{RH}_\infty,$$

where $\begin{bmatrix} -\hat{Y}_o \\ \hat{X}_o \end{bmatrix}$ is the right inverse of $\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}$. That is, it satisfies (9.4). It is obvious that

$$\begin{gathered}\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} \Delta_{\mathcal{K}} = \begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} - \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} \Longleftrightarrow \\ \begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} = \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} (I + \Delta_{\mathcal{K}}) .\end{gathered}$$

In the same manner, the result with the SIR can also be proved.

The following lemma is a known result, which provides us with the relation between $\Delta_{\mathcal{K}}$, $\Delta_{\mathcal{I}}$ in the model forms (9.8)–(9.9). The reference is given at the end of this chapter.

Lemma 9.2 *Given*

$$G_o(s) = \hat{M}_o^{-1}(s)\hat{N}_o(s) = N_o(s)M_o^{-1}(s), G(s) = N(s)M^{-1}(s)$$

with

$$\begin{bmatrix} M \\ N \end{bmatrix} = (I + \Delta_{\mathcal{I}}) \begin{bmatrix} M_o \\ N_o \end{bmatrix}, \Delta_{\mathcal{I}} \in \mathcal{RH}_\infty.$$

Then, it holds

$$\begin{bmatrix} \hat{M} & \hat{N} \end{bmatrix} = \begin{bmatrix} \hat{M}_o & \hat{N}_o \end{bmatrix} \left(I + \bar{\Delta}_{\mathcal{K}}\right)^{-1}, \tag{9.10}$$

$$\bar{\Delta}_{\mathcal{K}} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \Delta_{\mathcal{I}} \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \in \mathcal{RH}_\infty, \tag{9.11}$$

and $G(s) = \hat{M}^{-1}(s)\hat{N}(s)$ *is an LCF.*

It is straightforward that (9.10)–(9.11) can also be equivalently expressed by

$$\begin{bmatrix} \hat{M} & -\hat{N} \end{bmatrix} = \begin{bmatrix} \hat{M}_o & -\hat{N}_o \end{bmatrix} \left(I + \hat{\Delta}_{\mathcal{K}}\right)^{-1}, \tag{9.12}$$

$$\hat{\Delta}_{\mathcal{K}} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \bar{\Delta}_{\mathcal{K}} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \Delta_{\mathcal{I}} \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}. \tag{9.13}$$

It is worth noting that when

$$\Delta_{\mathcal{I}} = \begin{bmatrix} \delta_M & 0 \\ 0 & \delta_N \end{bmatrix},$$

it turns out

$$\hat{\Delta}_{\mathcal{K}} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \Delta_{\mathcal{I}} \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} = \begin{bmatrix} \delta_N & 0 \\ 0 & \delta_M \end{bmatrix}.$$

A direct application of this result is given in the following theorem.

Theorem 9.2 *Given* $G(s) = \hat{M}^{-1}(s)\hat{N}(s)$ *with*

$$\begin{gathered}\hat{M}(s) = \hat{M}_o(s) + \hat{M}_o(s)\delta_{\hat{M}}(s),\ \hat{N}(s) = \hat{N}_o(s) + \hat{N}_o(s)\delta_{\hat{N}}(s), \\ \delta_{\hat{M}}(s), \delta_{\hat{N}}(s) \in \mathcal{RH}_\infty,\ \left\|\delta_{\hat{M}}\right\|_\infty < 1,\ \left\|\delta_{\hat{N}}\right\|_\infty < 1,\end{gathered} \tag{9.14}$$

then

$$M(s) = \left(I + \delta_{\hat{N}}\right)^{-1} M_o(s) = \left(I + \delta_M\right) M_o(s), \tag{9.15}$$

$$N(s) = \left(I + \delta_{\hat{M}}\right)^{-1} N_o(s) = \left(I + \delta_N\right) N_o(s), \tag{9.16}$$

$$\delta_M = -\left(I + \delta_{\hat{N}}\right)^{-1} \delta_{\hat{N}},\ \delta_N = -\left(I + \delta_{\hat{M}}\right)^{-1} \delta_{\hat{M}},$$

build a RCF of $G(s) = N(s)M^{-1}(s)$.

Proof The proof is straightforward. In fact, (9.14) can be equivalently written as

$$\mathcal{K} = \begin{bmatrix} \hat{M}_o & -\hat{N}_o \end{bmatrix} \left(I + \begin{bmatrix} \delta_{\hat{M}} & 0 \\ 0 & \delta_{\hat{N}} \end{bmatrix} \right).$$

It follows from Lemma 9.2 that

$$\begin{bmatrix} M \\ N \end{bmatrix} = \left(I + \Delta_{\mathcal{I}}\right)^{-1} \begin{bmatrix} M_o \\ N_o \end{bmatrix},\ \Delta_{\mathcal{I}} = \begin{bmatrix} \delta_{\hat{N}} & 0 \\ 0 & \delta_{\hat{M}} \end{bmatrix}.$$

Since $\left\|\delta_{\hat{M}}\right\|_\infty < 1,\ \left\|\delta_{\hat{N}}\right\|_\infty < 1$, it turns out

$$I + \Delta_{\mathcal{I}} \in \mathcal{RH}_\infty,\ I + \Delta_{\mathcal{I}}^{-1} \in \mathcal{RH}_\infty.$$

As a result, (M, N) satisfying (9.15)–(9.16) is a RC pair of $G(s)$. Note that

$$\left(I + \Delta_{\mathcal{I}}\right)^{-1} = I - \left(I + \Delta_{\mathcal{I}}\right)^{-1} \Delta_{\mathcal{I}}.$$

Hence, (9.15)–(9.16) are proved.

Thanks to the equivalence between the different forms of model uncertainties and faults, we will, in our subsequent study, focus on those faults and uncertainties modelled in the form of the left coprime factor without loss of generality.

9.2 Observer-Based Fault Detection Schemes

In this section, we will present observer-based schemes for detecting multiplicative faults in LTI systems with model uncertainties, as modelled in the last section.

9.2.1 Basic Ideas and Major Tasks

We first briefly introduce the basic ideas and formulate the major tasks, which will then be solved in the subsequent sub-sections.

As described in Sect. 4.1, any stable residual generator can be represented by an SKR of the system under consideration and further parameterised, for instance, by

$$r = R \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix}, \tag{9.17}$$

where $(\hat{M}_o, \hat{N}_o)$ is the (possibly normalised) LC pair of the nominal plant model and $R(s)$ is a known stable post-filter. In our subsequent study, it is assumed that an FDF is applied as residual generator. According to Lemma 4.1, any FDF of the form

$$\begin{aligned} \dot{\hat{x}}(t) &= (A - LC)\,\hat{x}(t) + (B - LD)\,u(t) + Ly(t), \\ r(t) &= y(t) - \hat{y}(t) = y(t) - C\hat{x}(t) - Du(t) \end{aligned}$$

can be re-written as

$$\begin{aligned} \dot{\hat{x}}(t) &= (A - L_oC)\,\hat{x}(t) + (B - L_oD)\,u(t) + L_oy(t), \\ r(s) &= R(s)\left(y(s) - \hat{y}(s)\right), R(s) = I - C\left(sI - A + LC\right)^{-1}\left(L - L_o\right), \end{aligned}$$

where L_o is the observer gain matrix for the (possibly normalised) LC pair $(\hat{M}_o, \hat{N}_o)$. Note that $R^{-1}(s) \in \mathcal{RH}_\infty$ and

$$R^{-1}(s) = I - C\left(sI - A + L_oC\right)^{-1}\left(L_o - L\right).$$

Remark 9.1 *Recall that in the state space computation of the normalised $(\hat{M}_o, \hat{N}_o)$ given in Theorem 9.1, an output transformation with transformation matrix $\bar{\Gamma}^{-1/2}$ is needed. Considering that the assumption of the normalised LC pair $(\hat{M}_o, \hat{N}_o)$ is often irrelevant in the observer-based residual generator design, in which an additional output transformation is not included, we assume, in the sequel and for the sake of simplifying notation, the system matrices C and D are normalised by $\bar{\Gamma}^{-1/2}$.*

Let $(\hat{M}, \hat{N})$ denote the LC pair of the plant model with model uncertainties or/and faults. In this case,

$$y(s) = \hat{M}^{-1}(s)\hat{N}(s)u(s) \Longrightarrow \begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0.$$

As a result, the dynamics of the residual generator (9.17) is governed by

$$
\begin{aligned}
r &= R\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}\begin{bmatrix} u \\ y \end{bmatrix} = R\begin{bmatrix} \varDelta_{\hat{N}} & -\varDelta_{\hat{M}} \end{bmatrix}\begin{bmatrix} u \\ y \end{bmatrix}, \qquad (9.18)\\
\varDelta_{\hat{M}} &= \hat{M} - \hat{M}_o, \varDelta_{\hat{N}} = \hat{N} - \hat{N}_o.
\end{aligned}
$$

Next, we analyse the dynamics of the residual generator in the closed-loop configuration. We consider the standard feedback control configuration sketched in Fig. 9.1 with $G(s)$ as the plant model with uncertainties and/or faults, $K(s)$ as the feedback controller and v as the reference signal. Denote all stabilisation controllers by

$$
\begin{gathered}
K(s) = -U(s)V^{-1}(s) = -\hat{V}^{-1}(s)\,\hat{U}(s), \qquad (9.19)\\
\begin{bmatrix} \hat{V} & \hat{U} \end{bmatrix} = \begin{bmatrix} X_o - Q\hat{N}_o & Y_o + Q\hat{M}_o \end{bmatrix}, \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} \hat{Y}_o + M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix},
\end{gathered}
$$

where $X_o, Y_o, \hat{X}_o, \hat{Y}_o$ are $\mathcal{RH}_\infty$ matrices satisfying (9.4) with respect to the LC and RC pairs of the nominal plant model, and $Q(s) \in \mathcal{RH}_\infty$ is the parameterisation matrix.

Substituting the closed-loop dynamics

$$
\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} v
$$

into the residual dynamics leads to

$$
\begin{aligned}
r &= R\begin{bmatrix} \varDelta_{\hat{N}} & -\varDelta_{\hat{M}} \end{bmatrix}\begin{bmatrix} u \\ y \end{bmatrix}\\
&= R\begin{bmatrix} \varDelta_{\hat{N}} & -\varDelta_{\hat{M}} \end{bmatrix}\begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N}_o - \varDelta_{\hat{N}} & \hat{M}_o + \varDelta_{\hat{M}} \end{bmatrix}^{-1} \begin{bmatrix} \hat{V} \\ 0 \end{bmatrix} v. \qquad (9.20)
\end{aligned}
$$

By Bezout identity

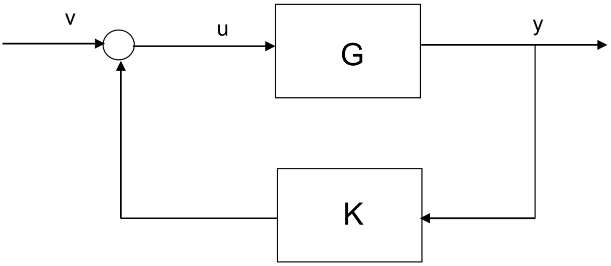


Fig. 9.1 Schematic description of a feedback control loop

$$\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N}_o & \hat{M}_o \end{bmatrix} = \begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N}_o & \hat{M}_o \end{bmatrix} \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} = I,$$

it is straightforward that

$$\begin{aligned} &\begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N}_o - \varDelta_{\hat{N}} & \hat{M}_o + \varDelta_{\hat{M}} \end{bmatrix}^{-1} \begin{bmatrix} \hat{V} \\ 0 \end{bmatrix} \\ &= \left(I + \begin{bmatrix} -U \\ V \end{bmatrix} \begin{bmatrix} -\varDelta_{\hat{N}} & \varDelta_{\hat{M}} \end{bmatrix} \right)^{-1} \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V}. \end{aligned}$$

It results in

$$r = R \begin{bmatrix} \varDelta_{\hat{N}} & -\varDelta_{\hat{M}} \end{bmatrix} \left(I + \begin{bmatrix} -U \\ V \end{bmatrix} \begin{bmatrix} -\varDelta_{\hat{N}} & \varDelta_{\hat{M}} \end{bmatrix} \right)^{-1} \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} \tag{9.21}$$

$$= -R \left(I + \begin{bmatrix} -\varDelta_{\hat{N}} & \varDelta_{\hat{M}} \end{bmatrix} \begin{bmatrix} -U \\ V \end{bmatrix} \right)^{-1} \begin{bmatrix} -\varDelta_{\hat{N}} & \varDelta_{\hat{M}} \end{bmatrix} \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v. \tag{9.22}$$

We would like to mention that in the above handling it is assumed that the residual generator and the LCF adopted in the controller parameterisation share the same observer gain matrix. Thanks to Lemma 4.1, this assumption loses no generality. It is evident that the stability of the observer-based FD system is guaranteed, as far as the feedback control system is stable. Also, the control performance of the closed-loop, expressed in terms of

$$\left(I + \begin{bmatrix} -\varDelta_{\hat{N}} & \varDelta_{\hat{M}} \end{bmatrix} \begin{bmatrix} -U \\ V \end{bmatrix} \right)^{-1},$$

has significant influence on the dynamics of the FD system.

Concerning the design of the observer-based fault detection system, threshold and the observer gain matrix should be determined. The threshold setting will be realised by determining

$$J_{th} = \sup_{\mathcal{K}_\varDelta, \hat{V}v \neq 0} \left(\frac{\|r\|_2}{\left\| \hat{V} v \right\|_2} \right) \left\| \hat{V} v \right\|_2$$

in the fault-free case, while the design of the observer gain matrix is performed based on the system dynamics both in the fault-free and faulty cases.

For the analysis of FD in open-loop configuration, let us substitute y by

$$y(s) = G(s)u(s) = \hat{M}^{-1}(s)\hat{N}(s)u(s) = N(s)M^{-1}(s)u(s)$$

into the residual dynamics (9.18), which leads to

$$r = R\left[-\hat{N}_o \;\; \hat{M}_o \right]\begin{bmatrix} u \\ y \end{bmatrix} = R\left(\hat{M}_o\hat{M}^{-1}\hat{N} - \hat{N}_o\right)u \tag{9.23}$$

$$= R\left[\Delta_{\hat{N}} \; -\Delta_{\hat{M}}\hat{M}^{-1} \right]\begin{bmatrix} I \\ \hat{N} \end{bmatrix}u. \tag{9.24}$$

Recall that the zeros of an LTI system are invariant with respect to the residual feedback. Thus, the dynamics of the FD system (9.23) is stable, only if the zeros of $\hat{M}$ in the RHP are also the RHP-zeros of $\hat{M}_o$. In the sequel, we assume that

$$\hat{M}_o\hat{M}^{-1} \in \mathcal{RH}_\infty,$$

and thus the observer-based FD system (9.23)–(9.24) is stable.

For the design of the observer-based FD system, the threshold will be determined according to

$$J_{th} = \sup_{\mathcal{K}_\Delta, u \neq 0}\left(\frac{\|r\|_2}{\|u\|_2}\right)\|u\|_2$$

in the fault-free case, and the observer gain matrix is to be found based on the system dynamics given in (9.24).

9.2.2 An FD System Design Scheme for Feedback Control Systems

Consider residual dynamics (9.21), which is re-written as

- fault-free but with uncertainty

$$r = -R\left(I + (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1}(\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o \\ N_o \end{bmatrix}\hat{V}v, \tag{9.25}$$

$$\mathcal{K}_\Delta = \left[-\hat{N}_\Delta(s) \;\; \hat{M}_\Delta(s) \right], \mathcal{K}_o = \left[-\hat{N}_o(s) \;\; \hat{M}_o(s) \right],$$

- faulty without considering uncertainty

$$r = -R\left(I + \left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1}\left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} M_o \\ N_o \end{bmatrix}\hat{V}v, \tag{9.26}$$

$$\mathcal{K}_f = \left[-\hat{N}_f(s) \;\; \hat{M}_f(s) \right].$$

It follows from Bezout identity and Lemma 9.1 that (9.26) can be further written into

$$\begin{aligned}r &= R\left(\mathcal{K}_f \begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \mathcal{K}_f \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v \\ &= R\left(\begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v.\end{aligned}$$

Remember that

$$R^{-1}(s) = I - C\left(sI - A + L_o C\right)^{-1}\left(L_o - L\right) \in \mathcal{RH}_\infty.$$

It turns out

$$\begin{aligned}r &= -R\left(I + \left(\mathcal{K}_\Delta - \mathcal{K}_o\right) \begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \left(\mathcal{K}_\Delta - \mathcal{K}_o\right) \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v \\ &= -\left(I + R\Delta_{\mathcal{K}} \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1}\right)^{-1} R \begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_{\mathcal{K}}\right) \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v, \qquad (9.27)\end{aligned}$$

$$\begin{aligned}r &= -R\left(\begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v \\ &= -\left(\begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1}\right)^{-1} \begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} M_o \\ N_o \end{bmatrix} \hat{V} v. \qquad (9.28)\end{aligned}$$

In (9.27) and (9.28), $\Delta_{\mathcal{K}}, \Delta_f$ are unknown and, in general, norm-bounded. On the assumption of a stable residual generator, it holds

$$\forall \omega, \sigma_{\max}\left(T\left(j\omega\right)\right) < \infty, T(s) = \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1},$$

which leads to

$$\forall \omega, \sigma_{\min}\left(T^{-1}\left(j\omega\right)\right) = \sigma_{\max}^{-1}\left(T\left(j\omega\right)\right) > 0,$$

where $\sigma_{\max}\left(T\left(j\omega\right)\right), \sigma_{\min}\left(T^{-1}\left(j\omega\right)\right)$ represent the maximum and minimum singular value of $T\left(j\omega\right), T^{-1}\left(j\omega\right)$, respectively. Define

$$\left\| \left(\begin{bmatrix} -\hat{N}_o \ \hat{M}_o \end{bmatrix} \left(I + \Delta_f\right) \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1}\right)^{-1} \right\|_- = \min_\omega \sigma_{\min}\left(T^{-1}\left(j\omega\right)\right).$$

It turns out

$$\left\| \left(\left[-\hat{N}_o \ \hat{M}_o \right] (I + \Delta_f) \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right)^{-1} \right\|_- = \frac{1}{\|T(s)\|_\infty}$$
$$\geq \frac{1}{\left\| \left[-\hat{N}_o \ \hat{M}_o \right] (I + \Delta_f) \right\|_\infty \left\| \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right\|_\infty}. \tag{9.29}$$

Moreover, it is a known result that

$$\left\| \left(I + R\Delta_{\mathcal{K}} \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right)^{-1} \right\|_\infty \leq \frac{1}{1 - \|R\Delta_{\mathcal{K}}\|_\infty \left\| \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right\|_\infty}. \tag{9.30}$$

The inequality (9.29) gives a lower-bound of

$$\left\| \left(\left[-\hat{N}_o \ \hat{M}_o \right] (I + \Delta_f) \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right)^{-1} \right\|_-,$$

which indicates the minimum influence of the fault on the residual dynamics. On the other hand, (9.30) provides us with an upper-bound for

$$\left\| \left(I + R\Delta_{\mathcal{K}} \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right)^{-1} \right\|_\infty,$$

which can be understood as the influence of the model uncertainties on the system stability. As a result of (9.29) and (9.30), reducing

$$\left\| \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right\|_\infty$$

leads to

- enhancing the minimum influence of the fault on r and thus increasing the fault detectability, and simultaneously,
- increasing the system robustness in the context of stability margin of the closed-loop system.

This motivates us to formulate the design of the observer-based residual generator as the optimisation problem

$$\min_L \left\| \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right\|_\infty. \tag{9.31}$$

Remark 9.2 *In the observer-based FD study on LTI systems, the so-called $\mathcal{H}_-/\mathcal{H}_\infty$ optimal design of the residual generator is a commonly adopted scheme, in which*

$\mathcal{H}_-$ and $\mathcal{H}_\infty$ indicate the minimum influence of the additive fault and the maximum influence of the (additive) disturbance on the residual, respectively, and the ratio $\mathcal{H}_-/\mathcal{H}_\infty$ is to be maximised. Our optimisation problem given in (9.31) can be interpreted as an extension of the $\mathcal{H}_-/\mathcal{H}_\infty$ optimal design scheme to the systems with multiplicative faults and model uncertainties, where

$$\left\| \left(\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} (I + \Delta_f) \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right)^{-1} \right\|_-$$

is the so-called $\mathcal{H}_-$-index.

Next, we study the optimisation problem (9.31). To this end, consider first the general form of $\begin{bmatrix} -U \\ V \end{bmatrix}$,

$$\begin{bmatrix} -U \\ V \end{bmatrix} = \begin{bmatrix} -\hat{Y}_o - M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix},$$

and its interpretation as Youla parameterisation of the controllers in the configuration in Fig. 5.1,

$$u = -UV^{-1}y.$$

Recall that in Sect. 5.2 it has been demonstrated that for $Q = 0$ the above controller can be written as

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + Lr(t), r(t) = y(t) - C\hat{x}(t) - Du(t), \\ u(t) &= F\hat{x}(t) \Longrightarrow \begin{bmatrix} u(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} -\hat{Y}_o(s) \\ \hat{X}_o(s) \end{bmatrix} r(s). \end{aligned}$$

A straightforward extension to the case $Q \neq 0$ yields

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} -\hat{Y}_o - M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix} r.$$

In fact, the controller or the SIR of the controller, $\begin{bmatrix} -U \\ V \end{bmatrix}$, is invariant to the observer gain matrix L, as illustrated by the following lemma.

Lemma 9.3 *Given*

$$\begin{aligned} \begin{bmatrix} -\hat{Y}_i(s) \\ \hat{X}_i(s) \end{bmatrix} &= \begin{bmatrix} 0 \\ I \end{bmatrix} + \begin{bmatrix} F \\ C_F \end{bmatrix} (sI - A_F)^{-1} L_i, i = 1, 2, \\ \begin{bmatrix} M_o(s) \\ N_o(s) \end{bmatrix} &= \begin{bmatrix} I \\ D \end{bmatrix} + \begin{bmatrix} F \\ C_F \end{bmatrix} (sI - A_F)^{-1} B, \\ Q_1(s) &\in \mathcal{RH}_\infty, A_F = A + BF, C_F = C + FD, \end{aligned}$$

then for

$$R(s) = I - C\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1) \in \mathcal{RH}_\infty,\ A_{L_2} = A - L_2C,$$

it holds

$$\left(\begin{bmatrix} -\hat{Y}_1 \\ \hat{X}_1 \end{bmatrix} - \begin{bmatrix} M_o \\ N_o \end{bmatrix} Q_1\right) = \left(\begin{bmatrix} -\hat{Y}_2 \\ \hat{X}_2 \end{bmatrix} - \begin{bmatrix} M_o \\ N_o \end{bmatrix} Q_2\right) R, \tag{9.32}$$

$$Q_2 = \left(Q_1 - F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\right) R^{-1} \tag{9.33}$$

$$= Q_1 R^{-1} - F\left(sI - A_{L_1}\right)^{-1}(L_1 - L_2). \tag{9.34}$$

Proof Consider

$$\hat{Y}_2(s)R(s) = -F(sI - A_F)^{-1}\left(L_1 + \left(I - L_2C\left(sI - A_{L_2}\right)^{-1}\right)(L_2 - L_1)\right).$$

Since

$$\begin{gathered} I - L_2C\left(sI - A_{L_2}\right)^{-1} = (sI - A)\left(sI - A_{L_2}\right)^{-1}, \\ F(sI - A_F)^{-1}(sI - A) = \left(I + F(sI - A_F)^{-1}B\right)F = M_o(s)F, \end{gathered}$$

it turns out

$$\hat{Y}_2(s)R(s) = \hat{Y}_1(s) - M_o(s)F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1). \tag{9.35}$$

Next, consider

$$\begin{aligned} \hat{X}_2(s)R(s) &= \left(I + C_F(sI - A_F)^{-1}L_2\right)\left(I - C\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\right) \\ &= \left(\hat{X}_1(s) + C_F(sI - A_F)^{-1}(L_2 - L_1)\right)\left(I - C\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\right). \end{aligned}$$

Note that

$$\begin{aligned} &C_F(sI - A_F)^{-1}(L_2 - L_1)\left(I - C\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\right) \\ &- \left(I + C_F(sI - A_F)^{-1}L_1\right)C\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1) \\ &= \left(C_F(sI - A_F)^{-1}(sI - A) - C\right)\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1) \\ &= N_o(s)F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1), \end{aligned}$$

which leads to

$$\hat{X}_2(s)R(s) = \hat{X}_1(s) + N_o(s)F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1). \tag{9.36}$$

Finally, the following straightforward computation,

$$\begin{aligned}
&F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\,R^{-1} \\
&= F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\left(I - C\left(sI - A_{L_1}\right)^{-1}(L_1 - L_2)\right) \\
&= F\left(sI - A_{L_1}\right)^{-1}(L_1 - L_2)\,,
\end{aligned}$$

yields

$$\left(Q_1 - F\left(sI - A_{L_2}\right)^{-1}(L_2 - L_1)\right)R^{-1} = Q_1 R^{-1} - F\left(sI - A_{L_1}\right)^{-1}(L_1 - L_2)\,.$$

As a result, it is evident that (9.32)–(9.34) are true.

Remark 9.3 *Equations (9.32)–(9.33) mean*

$$\begin{bmatrix} -\hat{Y}_1 - M_o Q_1 \\ \hat{X}_1 - N_o Q_1 \end{bmatrix} r_1 = \begin{bmatrix} -\hat{Y}_2 - M_o Q_2 \\ \hat{X}_2 - N_o Q_2 \end{bmatrix} r_2,\, r_2 = R r_1,$$

which is understood as the invariance of the controller with respect to the observer gain matrix L. In other words, it holds

$$\begin{aligned}
K(s) &= -\left(\hat{Y}_1 + M_o Q_1\right)\left(\hat{X}_1 - N_o Q_1\right)^{-1} \\
&= -\left(\hat{Y}_2 + M_o Q_2\right)\left(\hat{X}_2 - N_o Q_2\right)^{-1}, \\
Q_2 &= Q_1 R^{-1} - F\left(sI - A_{L_1}\right)^{-1}(L_1 - L_2) \in \mathcal{RH}_\infty, \\
R^{-1} &= I - C\left(sI - A_{L_1}\right)^{-1}(L_1 - L_2) \in \mathcal{RH}_\infty.
\end{aligned}$$

In the following example, we demonstrate how to apply Lemma 9.3 for determining the observer gain matrix for a given controller and the feedback control system configuration shown in Fig. 9.1.

Example 9.1 *Given the feedback control loop shown in Fig. 9.1 with an observer-based controller*

$$\begin{aligned}
u(t) &= F\hat{x}(t) + v(t), \\
\dot{\hat{x}}(t) &= (A - L_1 C)\,\hat{x}(t) + (B - L_1 D)\,u(t) + L_1 y(t),
\end{aligned} \tag{9.37}$$

where F, L_1 are given feedback control gain and observer gain, respectively. Our task is to find an observer-based residual generator

$$\begin{aligned}
\dot{\hat{x}}(t) &= (A - LC)\,\hat{x}(t) + (B - LD)\,u(t) + Ly(t), \\
r(t) &= y(t) - C\hat{x}(t) - Du(t),
\end{aligned} \tag{9.38}$$

so that the optimisation problem (9.31) is solved. Considering the invariance of the controller SIR with respect to the observer gain matrix, as demonstrated in Lemma 9.3, we can directly design the observer gain matrix L, as defined in (9.38), as follows:

- *Form*

$$\begin{bmatrix} -\hat{Y}_1 \\ \hat{X}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} + \begin{bmatrix} F \\ C_F \end{bmatrix} (sI - A_F)^{-1} L_1,$$

  where F, L_1 are given feedback control gain and observer gain, respectively;
- *Apply Lemma 9.3 to find the equivalent SIR of the controller as follows*

$$\begin{aligned} &\begin{bmatrix} -\hat{Y}_1 \\ \hat{X}_1 \end{bmatrix} R^{-1} = \begin{bmatrix} -\hat{Y} - MQ \\ \hat{X} - NQ \end{bmatrix}, \\ &\begin{bmatrix} -\hat{Y} \\ \hat{X} \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} + \begin{bmatrix} F \\ C_F \end{bmatrix} (sI - A_F)^{-1} L, \\ &Q = -F \left(sI - A_{L_1}\right)^{-1} (L_1 - L), A_{L_1} = A - L_1 C, \\ &R = I - C (sI - A_L)^{-1} (L - L_1), A_L = A - LC; \end{aligned}$$

- *Solve the optimisation problem*

$$\min_L \left\| \begin{bmatrix} -\hat{Y} - M_o Q \\ \hat{X} - N_o Q \end{bmatrix} \right\|_\infty .$$

  Note that $\begin{bmatrix} -\hat{Y} - M_o Q \\ \hat{X} - N_o Q \end{bmatrix}$ *can be written as*

$$\begin{aligned} &\begin{bmatrix} -\hat{Y} - M_o Q \\ \hat{X} - N_o Q \end{bmatrix} = T_1(s) + T_2(s)L, \\ &T_1 = \begin{bmatrix} 0 \\ I \end{bmatrix} + \left(\begin{bmatrix} I \\ D \end{bmatrix} + \begin{bmatrix} F \\ C_F \end{bmatrix} (sI - A_F)^{-1} B \right) F \left(sI - A_{L_1}\right)^{-1} L_1, \\ &T_2 = \begin{bmatrix} F \\ C_F \end{bmatrix} (sI - A_F)^{-1} \left(I - BF \left(sI - A_{L_1}\right)^{-1} \right) - \begin{bmatrix} I \\ D \end{bmatrix} F \left(sI - A_{L_1}\right)^{-1} . \end{aligned}$$

  Thus, we are able to solve the optimisation problem,

$$\min_L \left\| \begin{bmatrix} -\hat{Y} - M_o Q \\ \hat{X} - N_o Q \end{bmatrix} \right\|_\infty = \min_L \|T_1 + T_2 L\|_\infty ,$$

  for instance, using LMI (linear matrix inequality) technique.

Once the observer gain matrix L,

$$L = \arg\min_{L} \left\| \begin{bmatrix} -\hat{Y} - M_o Q \\ \hat{X} - N_o Q \end{bmatrix} \right\|_\infty ,$$

is determined, we can run the observer-based residual generator (9.38) or switch on the post-filter,

$$R = I - C \left(sI - A_L \right)^{-1} \left(L - L_1 \right),$$

to the residual generator that is embedded in the controller with the observer (9.37) as its core. It is of interest to notice that if the observer gain L_1 in the observer-based controller is set so that

$$L_1 = \arg\min_{L_1} \left\| \begin{bmatrix} -\hat{Y}_1 \\ \hat{X}_1 \end{bmatrix} \right\|_\infty ,$$

then

$$L = L_1$$

gives the optimal solution. In other words, we can directly use the observer embedded in the controller as an optimal residual generator.

As a summary of our study on the FD system design for feedback control systems, we claim that fault detectability and system robustness in the sense of stability margin can be consistently achieved by minimising the H_∞-norm of the SIR of the controller. That is,

$$\left\| \begin{bmatrix} -U \\ V \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} -\hat{Y}_o - M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix} \right\|_\infty$$

is minimised. In our subsequent investigation on fault-tolerant control issues in Chap. 19, we will reveal additional useful aspects of minimising the H_∞-norm of the SIR of a controller.

We would like to remark that our results in the above work can be generally formulated as: minimising

$$\left\| \begin{bmatrix} -U \\ V \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} -\hat{Y}_o - MQ \\ \hat{X}_o - NQ \end{bmatrix} \right\|_\infty$$

- increases the minimum influence of changes in the system on r, and simultaneously,
- enhances the system robustness in the context of stability margin, where changes in the system can be caused by faults or uncertainties.

Concerning the threshold setting, (9.25) is under consideration. Since multiplying a (non-zero) constant to the residual vector causes no change in the fault detectability, we assume, without loss of generality, that

$$\|R\|_\infty = 1,$$

where R is the post-filter of the observer-based residual generator (9.17). As a result, the threshold setting is achieved by solving the following optimisation problem

$$\gamma = \sup_{\mathcal{K}_\Delta - \mathcal{K}_o} \left\| R\left(I + (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o \\ N_o \end{bmatrix}\right\|_\infty ,$$

$$J_{th} = \gamma \left\| \hat{V} v \right\|_2 .$$

In our study, we assume $\mathcal{K}_\Delta - \mathcal{K}_o$ is unknown but norm-bounded by

$$\|\mathcal{K}_\Delta - \mathcal{K}_o\|_\infty \leq \delta_\Delta < 1.$$

It yields

$$\begin{aligned}
&\left\| \left(I + (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \right\|_\infty \leq \\
&\frac{1}{1 - \sup_{\|\mathcal{K}_\Delta - \mathcal{K}_o\|_\infty \leq \delta_\Delta} \left\| (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right\|_\infty} = \frac{1}{1 - \delta_\Delta b} \Longrightarrow \\
&\gamma = \sup_{\|\mathcal{K}_\Delta - \mathcal{K}_o\|_\infty \leq \delta_\Delta} \left\| R\left(I + (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o \\ N_o \end{bmatrix}\right\|_\infty \\
&= \frac{\delta_\Delta}{1 - \delta_\Delta b}, b = \left\| \begin{bmatrix} -U \\ V \end{bmatrix} \right\|_\infty \Longrightarrow \\
&J_{th} = \frac{\delta_\Delta}{1 - \delta_\Delta b} \left\| \hat{V} v \right\|_2 .
\end{aligned} \tag{9.39}$$

9.2.3 An FD System Design Scheme for Open-Loop Systems

Consider the residual generator (9.17) in the fault-free operation, which is further written as

$$r = R\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}\begin{bmatrix} u \\ y \end{bmatrix} = R\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}\begin{bmatrix} M_\Delta \\ N_\Delta \end{bmatrix} \nu$$

for some $\mathcal{L}_2$ bounded signal ν satisfying

$$u = M_\Delta \nu,$$

where $\begin{bmatrix} M_\Delta \\ N_\Delta \end{bmatrix}$ is the SIR of the uncertain plant. Using the uncertainty model

$$\begin{bmatrix} M_\Delta \\ N_\Delta \end{bmatrix} = (I + \Delta_{\mathcal{I}}) \begin{bmatrix} M_o \\ N_o \end{bmatrix}$$

results in

$$r = R \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} (I + \Delta_{\mathcal{I}}) \begin{bmatrix} M_o \\ N_o \end{bmatrix} v = R \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} \Delta_{\mathcal{I}} \begin{bmatrix} M_o \\ N_o \end{bmatrix} \nu.$$

Due to the uncertain $\Delta_{\mathcal{I}}$,

$$d := \Delta_{\mathcal{I}} \begin{bmatrix} M_o \\ N_o \end{bmatrix} \nu$$

is also a unknown vector. Hence,

$$r = R \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} d. \tag{9.40}$$

Recall our discussion on the unified solution and the fact that $\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}$ is normalised. As a result, the observer gain matrix L_o, as given in Theorem 9.1, delivers the optimal solution. That means, on the other hand,

$$R = I$$

is the optimal setting for the post-filter.

Next, we study the threshold setting issue. To this end, consider the dynamics of the residual generator (9.18) for $R = I$ and bring it into the form

$$\begin{aligned} r &= \begin{bmatrix} \Delta_{\hat{N}} & -\Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = \left(\Delta_{\hat{N}} - \Delta_{\hat{M}} \left(\hat{M}_o + \Delta_{\hat{M}} \right)^{-1} \left(\hat{N}_o + \Delta_{\hat{N}} \right) \right) u \\ &= \left(\Delta_{\hat{N}} - \Delta_{\hat{M}} \hat{M}_o^{-1} \left(I + \Delta_{\hat{M}} \hat{M}_o^{-1} \right)^{-1} \left(\hat{N}_o + \Delta_{\hat{N}} \right) \right) u. \end{aligned}$$

Remembering our discussion on the stability of observer-based residual generators for open-loop systems, we now assume, for the sake of simplicity,

$$\hat{M} \hat{M}_o^{-1} = \left(\hat{M}_o \hat{M}^{-1} \right)^{-1} \in \mathcal{RH}_\infty \Longrightarrow \Delta_{\hat{M}} \hat{M}_o^{-1} \in \mathcal{RH}_\infty.$$

Denote

$$\Delta_{\hat{M}} \hat{M}_o^{-1} = \bar{\Delta}_{\hat{M}},$$

and re-write the residual dynamics as

$$\begin{aligned} r &= \left(\Delta_{\hat{N}} - \Delta_{\hat{M}} \hat{M}_o^{-1} \left(I + \Delta_{\hat{M}} \hat{M}_o^{-1} \right)^{-1} \left(\hat{N}_o + \Delta_{\hat{N}} \right) \right) u \\ &= \left(\left(I + \bar{\Delta}_{\hat{M}} \right)^{-1} \Delta_{\hat{N}} - \bar{\Delta}_{\hat{M}} \left(I + \bar{\Delta}_{\hat{M}} \right)^{-1} \hat{N}_o \right) u. \end{aligned} \tag{9.41}$$

For our purpose, the following known inequality is useful.

Lemma 9.4 *Let $\Delta_1, \Delta_2 \in \mathcal{H}_\infty$ be such that*

$$\left\| \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} \right\|_\infty \leq b < 1.$$

Then,

$$\left\| \Delta_1 \left(I + \Delta_2 \right)^{-1} \right\|_\infty \leq \frac{b}{\sqrt{1 - b^2}}. \tag{9.42}$$

It is evident that the dual form of this lemma,

$$\left\| \begin{bmatrix} \Delta_1 & \Delta_2 \end{bmatrix} \right\|_\infty \leq b < 1 \Longrightarrow \left\| \left(I + \Delta_2 \right)^{-1} \Delta_1 \right\|_\infty \leq \frac{b}{\sqrt{1 - b^2}}, \tag{9.43}$$

also holds.

Now, suppose

$$\left\| \begin{bmatrix} \Delta_{\hat{N}} & \bar{\Delta}_{\hat{M}} \end{bmatrix} \right\|_\infty \leq \delta_{\bar{\Delta}} < 1.$$

It follows from (9.43) that

$$\left\| \left(I + \bar{\Delta}_{\hat{M}} \right)^{-1} \Delta_{\hat{N}} \right\|_\infty \leq \frac{\delta_{\bar{\Delta}}}{\sqrt{1 - \delta_{\bar{\Delta}}^2}}.$$

Note further

$$\left\| \left(I + \bar{\Delta}_{\hat{M}} \right)^{-1} \right\|_\infty \leq \frac{1}{1 - \delta_{\bar{\Delta}}} \Longrightarrow \left\| \bar{\Delta}_{\hat{M}} \left(I + \bar{\Delta}_{\hat{M}} \right)^{-1} \right\|_\infty \leq \frac{\delta_{\bar{\Delta}}}{1 - \delta_{\bar{\Delta}}}.$$

We have

$$\|r\|_2 \leq \frac{\delta_{\bar{\Delta}}}{\sqrt{1 - \delta_{\bar{\Delta}}^2}} \|u\|_2 + \frac{\delta_{\bar{\Delta}}}{1 - \delta_{\bar{\Delta}}} \left\| \hat{N}_o u \right\|_2,$$

which results in the threshold setting

$$J_{th} = \frac{\delta_{\bar{\Delta}}}{\sqrt{1-\delta_{\bar{\Delta}}^2}} \|u\|_2 + \frac{\delta_{\bar{\Delta}}}{1-\delta_{\bar{\Delta}}} \left\|\hat{N}_o u\right\|_2 \tag{9.44}$$

$$= \frac{\delta_{\bar{\Delta}}}{1-\delta_{\bar{\Delta}}} \left(\sqrt{\frac{1-\delta_{\bar{\Delta}}}{1+\delta_{\bar{\Delta}}}} \|u\|_2 + \left\|\hat{N}_o u\right\|_2 \right). \tag{9.45}$$

Note that the threshold (9.44) is a so-called adaptive threshold with online computation of $\|u\|_2$, $\left\|\hat{N}_o u\right\|_2$. In order to reduce online computations, we can, alternatively, set

$$J_{th} = \frac{\delta_{\bar{\Delta}}}{1-\delta_{\bar{\Delta}}} \left(\sqrt{\frac{1-\delta_{\bar{\Delta}}}{1+\delta_{\bar{\Delta}}}} + \left\|\hat{N}_o\right\|_\infty \right) \|u\|_2 . \tag{9.46}$$

9.3 System Analysis

In the observer-based fault detection and isolation (FDI) framework, control and observer theory as well as the associated design methods provide us with a powerful tool for the design of observer-based FDI systems. It is remarkable that system analysis plays an important role in control theory. For instance, the concept of stability margin is introduced to quantify how far a feedback control loop is from the instability. Although qualitative FDI performance evaluation has been addressed in some recent investigations, less attention has been, in comparison with the activities and efforts in control theory, devoted to this topic. In fact, few methods are available and applied for the analysis of FDI performance to give quantitative answers to those questions like how far a multiplicative fault is detectable, or how high the false alarm rate could become, or how far two different faults could be isolated. A quantisation of these features is helpful to get a deep insight into the system structural properties and thus for establishing appropriate design objectives. Analysis of FDI performance is of considerable practical interests.

The control theoretical tools applied for our investigation on FDI performance analysis are the coprime factorisation and gap metric techniques. Gap metric technique is widely applied in robust control theory for the stability analysis in uncertain closed-loops. Roughly speaking, a gap is a measurement of the distance between two closed subspaces in Hilbert space. The fact that the core of FDI study is to distinguish the influences of two variables/signals on the residuals, namely faults and disturbances/uncertainties for fault detection and two different faults in the fault isolation regard, motivates us to apply the gap metric technique to the analysis of fault diagnosis performance.

9.3.1 Graph and Gap Metrics

We first briefly review the gap metric technique. A good introduction to the gap metric technique can be found in the monographs by Vinnicombe and Feintuch. Let $\mathcal{H}_2$ denote the subspace of all signals which are of bounded energy and zero for $t < 0$. It follows from the SIR (5.4) of system

$$y = Gu = NM^{-1}u$$

that for $\nu \in \mathcal{H}_2$, all pairs (u, y) build a subspace in $\mathcal{H}_2$ and it is closed. This subspace is called the graph of the system and denoted by

$$\mathcal{G} = \left\{ z = \begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} M \\ N \end{bmatrix} \nu, \nu \in \mathcal{H}_2 \right\}. \tag{9.47}$$

Roughly speaking, a gap is a measurement of the distance between two closed subspaces in Hilbert space. Let $\mathcal{G}_1, \mathcal{G}_2$ be two graphs. The directed gap from $\mathcal{G}_1$ to $\mathcal{G}_2$, denoted by $\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2)$, is defined as

$$\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2) = \sup_{z_1 \in \mathcal{G}_1} \inf_{z_2 \in \mathcal{G}_2} \frac{\|z_1 - z_2\|_2}{\|z_1\|_2}. \tag{9.48}$$

It is clear that

$$0 \le \boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2) \le 1.$$

Let

$$G_1 = N_1 M_1^{-1}, G_2 = N_2 M_2^{-1}$$

be the normalised RCF of G_1, G_2, respectively. The directed gap $\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2)$ with

$$\mathcal{G}_i : \left\{ z_i = \begin{bmatrix} u_i \\ y_i \end{bmatrix} = \begin{bmatrix} M_i \\ N_i \end{bmatrix} \nu, \nu \in \mathcal{H}_2 \right\}, i = 1, 2,$$

can be calculated by solving the model matching problem (MMP)

$$\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2) = \inf_{Q \in \mathcal{H}_\infty} \left\| \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} - \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} Q \right\|_\infty. \tag{9.49}$$

The following two properties and results are known in the gap metric framework and widely used in robustness analysis of feedback control systems:

- Let $G_2 = \hat{M}_2^{-1} \hat{N}_2$ be the normalised LCF of G_2, then it holds

$$\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2) = \inf_{Q \in \mathcal{H}_\infty} \left\| \begin{bmatrix} M_2^* M_1 + N_2^* N_1 - Q \\ \hat{M}_2 N_1 - \hat{N}_2 M_1 \end{bmatrix} \right\|_\infty. \tag{9.50}$$

- Let $G = NM^{-1}$ be the normalised RCF of G and

$$G_1 = (N + \Delta_N)(M + \Delta_M)^{-1}, \Delta_N, \Delta_M \in \mathcal{H}_\infty,$$

then for all $0 < b \leq 1$

$$\{G_1 : \boldsymbol{\delta}(\mathcal{G}, \mathcal{G}_1) < b\} = \left\{ G_1 : \left\| \begin{bmatrix} \Delta_M \\ \Delta_N \end{bmatrix} \right\|_\infty < b \right\}. \tag{9.51}$$

The gap metric between $\mathcal{G}_1$ and $\mathcal{G}_2$ is defined by

$$\delta(\mathcal{G}_1, \mathcal{G}_2) = \max\{\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2), \boldsymbol{\delta}(\mathcal{G}_2, \mathcal{G}_1)\}. \tag{9.52}$$

Moreover, if $\delta(\mathcal{G}_1, \mathcal{G}_2) < 1$, then

$$\boldsymbol{\delta}(\mathcal{G}_1, \mathcal{G}_2) = \boldsymbol{\delta}(\mathcal{G}_2, \mathcal{G}_1) = \delta(\mathcal{G}_1, \mathcal{G}_2). \tag{9.53}$$

In robust control theory, the so-called ν-gap metric is often applied instead of the gap given in (9.49). We adopt the definition given by Vinnicombe.

Given

$$G_1 = N_1 M_1^{-1}, G_2 = \hat{M}_2^{-1} \hat{N}_2$$

being the normalised RCF of G_1 and LCF of G_2, respectively, then the ν-gap metric is defined as

$$\delta_\nu(\mathcal{G}_1, \mathcal{G}_2) = \begin{cases} \|\mathcal{K}_2 \mathcal{G}_1\|_\infty, & \text{if } \det(\mathcal{K}_2 \mathcal{G}_1)(j\omega) \neq 0, \omega \in (-\infty, \infty) \\ & \text{and } wno(\det(\mathcal{K}_2 \mathcal{G}_1)) = 0, \\ 1, & \text{otherwise,} \end{cases} \tag{9.54}$$

where $wno(\det(\mathcal{K}_2 \mathcal{G}_1))$ denotes the winding number about the origin of $\det(\mathcal{K}_2 \mathcal{G}_1)$, and

$$\mathcal{K}_2 \mathcal{G}_1 = \hat{M}_2 N_1 - \hat{N}_2 M_1.$$

Remark 9.4 *In the book by Vinnicombe, the winding number is defined and well described. Since it is not directly used in our work, we will not go into details on* $wno(\det(\mathcal{K}_2 \mathcal{G}_1))$.

It is of interest to notice that

- according to (9.50), it holds in general

$$\delta_\nu(\mathcal{G}_1, \mathcal{G}_2) \leq \delta(\mathcal{G}_1, \mathcal{G}_2),$$

and, in this regard, the ν-gap metric is said to be less conservative than gap metric;
- $\delta_\nu(\mathcal{G}_1, \mathcal{G}_2)$ is a metric and hence

$$\delta_\nu\left(\mathcal{G}_1, \mathcal{G}_2\right) = \delta_\nu\left(\mathcal{G}_2, \mathcal{G}_1\right).$$

We now remove the winding number condition in the definition (9.54) of $\delta_\nu\left(\mathcal{G}_1, \mathcal{G}_2\right)$, which results in the so-called $\mathcal{L}_2$-gap metric. For our purpose, we adopt the following definition introduced in the book by Vinnicombe.

Definition 9.1 *Given*

$$G_1 = N_1 M_1^{-1}, G_2 = N_2 M_2^{-1}$$

being the normalised RCFs of G_1 and G_2, respectively, then the $\mathcal{L}_2$-gap metric is defined as

$$\delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) = \inf_{Q \in \mathcal{L}_\infty} \left\| \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} - \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} Q \right\|_\infty . \tag{9.55}$$

Let $G_2 = \hat{M}_2^{-1} \hat{N}_2$ be the normalised LCF of G_2. Since

$$\begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \\ M_2^* & N_2^* \end{bmatrix}^* \begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \\ M_2^* & N_2^* \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix},$$
$$\begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \\ M_2^* & N_2^* \end{bmatrix} \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \begin{bmatrix} M_2^* & N_2^* \end{bmatrix} \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} \in \mathcal{L}_\infty,$$

it turns out

$$\begin{aligned} \delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) &= \inf_{Q \in \mathcal{L}_\infty} \left\| \begin{bmatrix} \hat{M}_2 N_1 - \hat{N}_2 M_1 \\ M_2^* M_1 + N_2^* N_1 - Q \end{bmatrix} \right\|_\infty \\ &= \left\| \begin{bmatrix} \hat{M}_2 N_1 - \hat{N}_2 M_1 \end{bmatrix} \right\|_\infty = \left\| \mathcal{K}_2 \mathcal{G}_1 \right\|_\infty . \end{aligned} \tag{9.56}$$

As a result, we have the following relationship between the gap, ν-gap and $\mathcal{L}_2$-gap metrics:

$$\delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) \le \delta_\nu\left(\mathcal{G}_1, \mathcal{G}_2\right) \le \delta\left(\mathcal{G}_1, \mathcal{G}_2\right). \tag{9.57}$$

In fact, it should be proved that $\delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right)$ given in the above definition is a metric. This is given in the following theorem.

Theorem 9.3 $\delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right)$ *defined in Definition 9.1 satisfies*

$$\begin{gathered} (i)\ \delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) \ge 0, \delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) = 0 \ if\ and\ only\ if\ \mathcal{G}_1 = \mathcal{G}_2 \\ (ii)\ \delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) = \delta_{\mathcal{L}_2}\left(\mathcal{G}_2, \mathcal{G}_1\right). \end{gathered}$$

Moreover, it holds, for $\delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right)$, $\delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_3\right)$, $\delta_{\mathcal{L}_2}\left(\mathcal{G}_2, \mathcal{G}_3\right)$ defined in Definition 9.1,

$$(iii)\ \delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_2\right) \le \delta_{\mathcal{L}_2}\left(\mathcal{G}_1, \mathcal{G}_3\right) + \delta_{\mathcal{L}_2}\left(\mathcal{G}_2, \mathcal{G}_3\right).$$

The proof of this theorem can be done along the lines in the proof of Theorem 3.1 in the book by Vinnicombe.

With the properties (i)-(iii) given above, $\delta_{\mathcal{L}_2}(\mathcal{G}_1, \mathcal{G}_2)$ is a metric.

Remark 9.5 *In his book, Vinnicombe has pointed out that $\mathcal{L}_2$-gap metric is, different from the ν-gap metric, useless in dealing with feedback control systems. In our subsequent work on FDI performance analysis, it seems that $\mathcal{L}_2$-gap metric is a useful indicator.*

9.3.2 The $\mathcal{K}$-gap

In the literature (see the references given at the end of the chapter), the so-called (directed) T-gap is introduced using the graph

$$\text{graph}\left(G^T\right) = \begin{bmatrix} \hat{M}^T(s) \\ \hat{N}^T(s) \end{bmatrix} \mathcal{H}_2, \, G(s) = \hat{M}^{-1}(s)\hat{N}(s),$$

and defined by

$$\begin{aligned} &\boldsymbol{\delta}_T\left(\mathcal{G}_1, \mathcal{G}_2\right) \\ &= \inf_{Q \in \mathcal{H}_\infty} \left\| \begin{bmatrix} \hat{M}_1(s) & \hat{N}_1(s) \end{bmatrix} - Q \begin{bmatrix} \hat{M}_2(s) & \hat{N}_2(s) \end{bmatrix} \right\|_\infty . \end{aligned} \tag{9.58}$$

Note that

$$\left\| \begin{bmatrix} \hat{M}_1(s) & \hat{N}_1(s) \end{bmatrix} - Q \begin{bmatrix} \hat{M}_2(s) & \hat{N}_2(s) \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} \hat{M}_1^T(s) \\ \hat{N}_1^T(s) \end{bmatrix} - \begin{bmatrix} \hat{M}_2^T(s) \\ \hat{N}_2^T(s) \end{bmatrix} Q^T \right\|_\infty .$$

Hence, the T-gap is a direct extension of $\boldsymbol{\delta}\left(\mathcal{G}_1, \mathcal{G}_2\right)$ defined in (9.49).

Below, we introduce a new gap definition, the so-called $\mathcal{K}$-gap. Although the $\mathcal{K}$-gap is, from the computation point of view, equivalent to the T-gap, it is introduced and defined as a measurement of the distance between two kernel subspaces, which will serve as a tool for our system analysis and synthesis in the FDI framework. For our purpose, the following graph definition is introduced

$$\mathcal{K} = \left\{ \begin{bmatrix} u \\ y \end{bmatrix} : \begin{bmatrix} -\hat{N}(s) & \hat{M}(s) \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0, \begin{bmatrix} u \\ y \end{bmatrix} \in \mathcal{H}_2 \right\}, \tag{9.59}$$

which represents the subspace of $\mathcal{H}_2 \times \mathcal{H}_2$ consisting of all input and output pairs (u, y) satisfying

$$\begin{bmatrix} \hat{M}(s) & -\hat{N}(s) \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0.$$

It is a closed subspace in $\mathcal{H}_2$. It is worth mentioning that the graph can be equivalently defined using any SKR of G. To measure the distance between two different graphs $\mathcal{K}_1$ and $\mathcal{K}_2$, in light of the definition of a directed gap (9.48), we now introduce the following definition.

Definition 9.2 *Let $\hat{M}_i^{-1}(s)\hat{N}_i(s)$ be an LCF of $G_i(s)$, and*

$$\mathcal{K}_i = \left\{ \begin{bmatrix} u_i \\ y_i \end{bmatrix} : \begin{bmatrix} -\hat{N}_i(s) & \hat{M}_i(s) \end{bmatrix} \begin{bmatrix} u_i \\ y_i \end{bmatrix} = 0, \begin{bmatrix} u_i \\ y_i \end{bmatrix} \in \mathcal{H}_2 \right\} i = 1, 2. \tag{9.60}$$

We call

$$\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_1, \mathcal{K}_2\right) = \sup_{\begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \in \mathcal{K}_1} \inf_{\begin{bmatrix} u_2 \\ y_2 \end{bmatrix} \in \mathcal{K}_2} \frac{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2} \tag{9.61}$$

directed $\mathcal{K}$-gap.

Roughly speaking, $\mathcal{K}$-gap (9.61) is a measurement of the directed distance between two kernel subspaces in $\mathcal{H}_2$, which are spanned by input and output signals from two different processes, respectively. In the FDI context, these two sets of input and output signals are understood as input signals of two kernel representations. Moreover, it is worth noticing that unlike gap metric, we are often more interested in the directed $\mathcal{K}$-gap. This is due to the fact that from the residual generation perspective, a quantisation answer of the distance from the nominal plant to the uncertain/faulty system is of more significance.

Next, in line with the existing results, for instance the ones reported in the paper by Georgiou (see the reference at the end of the chapter), we study the computation scheme of the above introduced $\mathcal{K}$-gap.

Theorem 9.4 *Given $\mathcal{K}_i, i = 1, 2,$ as defined in Definition 9.2 with $\hat{M}_i(s)$, $\hat{N}_i(s)$ being the normalised LCF, then it holds*

$$\begin{aligned} &\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_1, \mathcal{K}_2\right) \\ &= \inf_{Q \in \mathcal{H}_\infty} \left\| \begin{bmatrix} -\hat{N}_1(s) & \hat{M}_1(s) \end{bmatrix} - Q \begin{bmatrix} -\hat{N}_2(s) & \hat{M}_2(s) \end{bmatrix} \right\|_\infty . \end{aligned} \tag{9.62}$$

Proof Let $\mathcal{K}_i^\perp$ be the orthogonal complement of subspace $\mathcal{K}_i$, and $\Pi_{\mathcal{K}_i}$, $\Pi_{\mathcal{K}_i^\perp}$ be the orthogonal projection onto $\mathcal{K}_i, \mathcal{K}_i^\perp$, respectively. Recall that $\hat{M}_i(s)$ and $\hat{N}_i(s)$ build the normalised left coprime pair, that is

$$\begin{aligned} &\begin{bmatrix} -\hat{N}_i(s) & \hat{M}_i(s) \end{bmatrix} \begin{bmatrix} -\hat{N}_i(s) & \hat{M}_i(s) \end{bmatrix}^* \\ &= \begin{bmatrix} \hat{N}_i(s) & \hat{M}_i(s) \end{bmatrix} \begin{bmatrix} \hat{N}_i(s) & \hat{M}_i(s) \end{bmatrix}^* = I. \end{aligned}$$

It can be proved that

$$\Pi_{\mathcal{K}_i} = I - \left[-\hat{N}_i(s)\ \hat{M}_i(s) \right]^* \left[-\hat{N}_i(s)\ \hat{M}_i(s) \right], \tag{9.63}$$

$$\Pi_{\mathcal{K}_i^\perp} = \left[-\hat{N}_i(s)\ \hat{M}_i(s) \right]^* \left[-\hat{N}_i(s)\ \hat{M}_i(s) \right]. \tag{9.64}$$

Since

$$\begin{bmatrix} u_1 \\ y_1 \end{bmatrix} = \Pi_{\mathcal{K}_2} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} + \Pi_{\mathcal{K}_2^\perp} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix},$$

we have

$$\inf_{\begin{bmatrix} u_2 \\ y_2 \end{bmatrix} \in \mathcal{K}_2} \frac{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2} = \frac{\left\| \Pi_{\mathcal{K}_2^\perp} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2},$$

which yields

$$\boldsymbol{\delta}_K\left(\mathcal{K}_1, \mathcal{K}_2\right) = \sup_{\begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \in \mathcal{K}_1} \frac{\left\| \Pi_{\mathcal{K}_2^\perp} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2} =: \left\| \Pi_{\mathcal{K}_2^\perp} \Pi_{\mathcal{K}_1} \right\|.$$

Here, $\left\| \Pi_{\mathcal{K}_2^\perp} \Pi_{\mathcal{K}_1} \right\|$ denotes a norm of the operator $\Pi_{\mathcal{K}_2^\perp} \Pi_{\mathcal{K}_1}$. On account of (9.63) and (9.64), it turns out

$$\begin{aligned} &\left\| \Pi_{\mathcal{K}_2^\perp} \Pi_{\mathcal{K}_1} \right\| = \left\| \left(\Pi_{\mathcal{K}_2^\perp} \Pi_{\mathcal{K}_1} \right)^T \right\| \\ &= \left\| \left(I - \mathcal{K}_1^T \left(\mathcal{K}_1^T\right)^* \right) \mathcal{K}_2^T \left(\mathcal{K}_2^T\right)^* \right\| = \left\| \left(I - \mathcal{K}_1^T \left(\mathcal{K}_1^T\right)^* \right) \mathcal{K}_2^T \right\| \\ &\mathcal{K}_i = \left[-\hat{N}_i(s)\ \ \hat{M}_i(s) \right], i = 1, 2. \end{aligned}$$

In the paper by Georgiou (see the reference in the end of this chapter), it has been pointed out that using the commutant lifting theorem, it can be proved that

$$\begin{aligned} &\left\| \left(I - \mathcal{K}_1^T \left(\mathcal{K}_1^T\right)^* \right) \mathcal{K}_2^T \right\| \\ &= \inf_{Q^T \in \mathcal{H}_\infty} \left\| \begin{bmatrix} -\hat{N}_1^T(s) \\ \hat{M}_1^T(s) \end{bmatrix} - \begin{bmatrix} -\hat{N}_2^T(s) \\ \hat{M}_2^T(s) \end{bmatrix} Q^T \right\|_\infty \\ &= \inf_{Q \in \mathcal{H}_\infty} \left\| \left[-\hat{N}_1(s)\ \ \hat{M}_1(s) \right] - Q \left[-\hat{N}_2(s)\ \ \hat{M}_2(s) \right] \right\|_\infty, \end{aligned}$$

which completes the proof.

It is straightforward from the $\mathcal{K}$-gap computation that

$$0 \le \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_1, \mathcal{K}_2\right) \le 1.$$

Let

$$G_2(s) = N_2(s) M_2^{-1}(s)$$

be the normalised RCF of $G_2(s)$, then we have

$$\begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \\ M_2^* & N_2^* \end{bmatrix} \begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \\ M_2^* & N_2^* \end{bmatrix}^* = I.$$

It is easy to see

$$\begin{aligned} &\left\| \left(\begin{bmatrix} -\hat{N}_1 & \hat{M}_1 \end{bmatrix} - Q \begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \end{bmatrix} \right) \right\|_\infty \\ &= \left\| \left(\begin{bmatrix} -\hat{N}_1 & \hat{M}_1 \end{bmatrix} - Q \begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \end{bmatrix} \right) \begin{bmatrix} -\hat{N}_2^* & M_2 \\ \hat{M}_2^* & N_2 \end{bmatrix} \right\|_\infty \\ &= \left\| \begin{bmatrix} \hat{M}_1 \hat{M}_2^* + \hat{N}_1 \hat{N}_2^* - Q & \hat{M}_1 N_2 - \hat{N}_1 M_2 \end{bmatrix} \right\|_\infty . \end{aligned}$$

As a result, we present the following corollary, which can be applied as an approximation of the $\mathcal{K}$-gap computation as given in (9.62).

Corollary 9.1 *Let $\hat{M}_1(s)$, $\hat{N}_1(s)$ be the normalised LCF of $G_1(s)$, and $M_2(s)$, $N_2(s)$ be the normalised RCF of $G_2(s)$, then it holds that*

$$\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_1, \mathcal{K}_2\right) \ge \left\| \hat{M}_1 N_2 - \hat{N}_1 M_2 \right\|_\infty .$$

It is of interest to note that analogue to Definition 9.1, we can define

$$\delta_{\mathcal{L}_2} \left(\mathcal{K}_1, \mathcal{K}_2\right) := \inf_{Q \in \mathcal{L}_\infty} \left\| \begin{bmatrix} -\hat{N}_1 & \hat{M}_1 \end{bmatrix} - Q \begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \end{bmatrix} \right\|_\infty , \tag{9.65}$$

which results in

$$\delta_{\mathcal{L}_2} \left(\mathcal{K}_1, \mathcal{K}_2\right) = \left\| \hat{M}_1 N_2 - \hat{N}_1 M_2 \right\|_\infty .$$

Recall, on the other hand, that

$$\begin{gathered} \delta_{\mathcal{L}_2} \left(\mathcal{G}_1, \mathcal{G}_2\right) = \delta_{\mathcal{L}_2} \left(\mathcal{G}_2, \mathcal{G}_1\right) \Longrightarrow \\ \left\| \left[\hat{M}_2 N_1 - \hat{N}_2 M_1 \right] \right\|_\infty = \left\| \hat{M}_1 N_2 - \hat{N}_1 M_2 \right\|_\infty . \end{gathered}$$

As a result,

$$\delta_{\mathcal{L}_2} \left(\mathcal{K}_1, \mathcal{K}_2\right) = \delta_{\mathcal{L}_2} \left(\mathcal{G}_1, \mathcal{G}_2\right). \tag{9.66}$$

Thus, in our subsequent investigations, we use both notations, $\delta_{\mathcal{L}_2} \left(\mathcal{G}_1, \mathcal{G}_2\right)$ and $\delta_{\mathcal{L}_2} \left(\mathcal{K}_1, \mathcal{K}_2\right)$, for the $\mathcal{L}_2$-gap metric.

It is worth mentioning that although they have been derived in different ways, the (directed) $\mathcal{K}$-gap and T-gap are equivalent. This can be easily seen from the fact that

$$\begin{aligned}\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1,\mathcal{K}_2\right) &= \inf_{Q\in\mathcal{H}_\infty}\left\|\begin{bmatrix}-\hat{N}_1 & \hat{M}_1\end{bmatrix} - Q\begin{bmatrix}-\hat{N}_2 & \hat{M}_2\end{bmatrix}\right\|_\infty \\ &= \inf_{Q\in\mathcal{H}_\infty}\left\|\left(\begin{bmatrix}-\hat{N}_1 & \hat{M}_1\end{bmatrix} - Q\begin{bmatrix}-\hat{N}_2 & \hat{M}_2\end{bmatrix}\right)\begin{bmatrix}0 & -I \\ I & 0\end{bmatrix}\right\|_\infty \\ &= \inf_{Q\in\mathcal{H}_\infty}\left\|\begin{bmatrix}\hat{M}_1 & \hat{N}_1\end{bmatrix} - Q\begin{bmatrix}\hat{M}_2 & \hat{N}_2\end{bmatrix}\right\|_\infty = \boldsymbol{\delta}_T\left(\mathcal{G}_1,\mathcal{G}_2\right).\end{aligned}$$

At the end of this sub-section, we recall the fact that any (stable) FDF can be parameterised by

$$r(s) = R(s)\left(-\hat{N}_o(s)u(s) + \hat{M}_o(s)y(s)\right),$$

where the post-filter $R(s) \in \mathcal{RH}_\infty$ is the parameterisation system with $R^{-1}(s) \in \mathcal{RH}_\infty$, and $(\hat{M}_o, \hat{N}_o)$ is a normalised LC pair. This allows us to substitute the normalised LC pair $(\hat{M}_2, \hat{N}_2)$ in the computation of $\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1,\mathcal{K}_2\right)$ and $\delta_{\mathcal{L}_2}\left(\mathcal{K}_1,\mathcal{K}_2\right)$, as given in (9.62) and (9.65) respectively, by any (not necessarily normalised) LC pair. Note that a consequence of this extension is that if

$$\exists R(s) \in \mathcal{RH}_\infty, \mathcal{K}_2 = \begin{bmatrix}-\hat{N}_2 & \hat{M}_2\end{bmatrix} = R\begin{bmatrix}-\hat{N}_1 & \hat{M}_1\end{bmatrix} = R\mathcal{K}_1,$$

then we have

$$\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1,\mathcal{K}_2\right) = 0, \delta_{\mathcal{L}_2}\left(\mathcal{K}_1,\mathcal{K}_2\right) = 0.$$

This can be interpreted as the transfer function G_2 being identical with G_1. It is worth mentioning that this extension of the computation of $\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1,\mathcal{K}_2\right)$ and $\delta_{\mathcal{L}_2}\left(\mathcal{K}_1,\mathcal{K}_2\right)$ makes sense from the fault detection point of view. For instance, when

$$\exists R(s) \in \mathcal{RH}_\infty \text{ s.t. } \mathcal{K}_f = R\mathcal{K}_o \Longrightarrow \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o,\mathcal{K}_f\right) = 0,$$

then the fault $\mathcal{K}_f$ cannot be detected.

9.3.3 Residual Dynamics with Respect to Model Uncertainties in Feedback Control Systems

As discussed in the previous sub-section, the $\mathcal{K}$-gap provides us with quantisation measures of how far the kernel subspaces of two systems are. This fact motivates us to apply $\mathcal{K}$-gap as an indicator for the distance from the nominal system (9.1) to the uncertain model (9.7) in the closed-loop configuration presented in Fig. 9.1, and associated with it, to study the influence of model uncertainties on the dynamics of residual generators.

Consider residual generator (9.17) with $\|R(s)\|_\infty = \beta(> 0)$ and recall

$$r = -R\left(I + (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1}(\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o \\ N_o \end{bmatrix}\hat{V}v,$$
$$\mathcal{K}_\Delta = \begin{bmatrix} -\hat{N}_\Delta & \hat{M}_\Delta \end{bmatrix}, \mathcal{K}_o = \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}.$$

It is a well-known result in robust control theory that the set of unstructured model uncertainties can be equivalently characterised by the gap metric. Analogue to this result, we give the following lemma without proof.

Lemma 9.5 *Given nominal and uncertain system models G_o and G_Δ with SKRs $\mathcal{K}_o$ and $\mathcal{K}_\Delta$, and let $0 \leq \delta_\Delta < 1$, then it holds*

$$\{G_\Delta : \boldsymbol{\delta}_\mathcal{K}(\mathcal{K}_o, \mathcal{K}_\Delta) < \delta_\Delta\} = \{G_\Delta : \|\mathcal{K}_\Delta - \mathcal{K}_o\|_\infty < \delta_\Delta\}. \tag{9.67}$$

Next, we study the residual dynamics from a different aspect than the discussion in the previous sub-sections, in order to gain a deeper insight into fault detection in the closed-loop configuration.

Recall that

$$\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} = \begin{bmatrix} M_o & -\hat{Y}_o \\ N_o & \hat{X}_o \end{bmatrix}\begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix}$$

is the composition of the SIRs of the controller and the (nominal) plant model, and thus contains full information of the closed-loop dynamics. Note further

$$\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} = \begin{bmatrix} 0 & I \end{bmatrix},$$
$$\inf_{\bar{Q}\in\mathcal{H}_\infty}\left\|\left(\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix} - \bar{Q}\begin{bmatrix} -\hat{N}_\Delta & \hat{M}_\Delta \end{bmatrix}\right)\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}\right\|_\infty \in [0, 1].$$

This motivates us to introduce the concept of $\mathcal{K}$-gap of closed-loop systems.

Definition 9.3 *Given a feedback control system as shown in Fig. 9.1 with*

$$\begin{bmatrix} \mathcal{I}_o & \mathcal{I}_u \end{bmatrix} = \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}, \begin{bmatrix} \mathcal{K}_o \\ \mathcal{K}_\Delta \end{bmatrix} = \begin{bmatrix} -\hat{N}_o & \hat{M}_o \\ -\hat{N}_\Delta & \hat{M}_\Delta \end{bmatrix}$$

as the SIRs of the nominal model and controller as well as the SKRs of the nominal and uncertain models, respectively,

$$\mathcal{K}_o^{cl} = \begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}, \mathcal{K}_\Delta^{cl} = \begin{bmatrix} -\hat{N}_\Delta & \hat{M}_\Delta \end{bmatrix}\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \tag{9.68}$$

are called SKRs of the nominal and uncertain closed-loop model, respectively, and

$$\delta_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_\Delta^{cl}\right) = \inf_{\bar{Q}\in\mathcal{H}_\infty} \left\|\mathcal{K}_o^{cl} - \bar{Q}\mathcal{K}_\Delta^{cl}\right\|_\infty \tag{9.69}$$

is called $\mathcal{K}$-gap of closed-loop system.

In other words,

$$\mathcal{K}_\Delta^{cl} - \mathcal{K}_o^{cl} = (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \tag{9.70}$$

can be interpreted as a representation of the difference between the nominal and real plant SKRs in the closed-loop configuration.

Theorem 9.5 *Given the closed-loop SKRs $\mathcal{K}_\Delta^{cl}, \mathcal{K}_o^{cl}$ satisfying*

$$\delta_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_\Delta^{cl}\right) \leq \delta_\Delta^{cl} < 1,$$

then it holds for the residual generator (9.17),

$$\|r\|_2 \leq \frac{\beta\delta_\Delta^{cl}}{\sqrt{1-\left(\delta_\Delta^{cl}\right)^2}}\bar{w}, \bar{w} = \left\|\hat{V}v\right\|_2. \tag{9.71}$$

Proof Since

$$\left\|(\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}\right\|_\infty = \left\|\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}^T (\mathcal{K}_\Delta - \mathcal{K}_o)^T\right\|_\infty,$$

$$r = R\left(I + (\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1}(\mathcal{K}_\Delta - \mathcal{K}_o)\begin{bmatrix} M_o \\ N_o \end{bmatrix}\hat{V}v,$$

and noting the relation (9.67), (9.71) follows immediately from Lemma 9.4.

In comparison with the threshold setting rule given in (9.39), as the upper bound of the $\mathcal{L}_2$-norm of the residual, (9.71) is less conservative and compactly expressed in terms of the $\mathcal{K}$ -gap between $\mathcal{K}_\Delta^{cl}$ and $\mathcal{K}_o^{cl}$.

Example 9.2 *As an illustrating example for the above results, we consider such uncertain systems, which are described by*

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) + E_\eta\eta(t), \eta(t) \in \mathcal{R}^{p_\eta}, \\
y(t) &= Cx(t) + Du(t) + F_\eta\eta(t), \\
\gamma(t) &= C_\gamma x(t) + D_\gamma u(t) + F_\gamma\eta(t) \in \mathcal{R}^{m_\gamma}, \\
\eta(s) &= \Delta(s)\gamma(s), \Delta(s) \in \mathcal{RH}_\infty,
\end{aligned}$$

where $\Delta(s)$ represents the uncertainty, and matrices $E_\eta, F_\eta, C_\gamma, D_\gamma$ and F_γ are known and of appropriate dimensions. It is known that the linear fractional transformation (LFT) model of this system, $y(s) = G_\Delta(s)u(s)$, is

$$\begin{aligned} G_\Delta(s) &= G_{11}(s) + G_{12}(s)\Delta(s)\left(I - G_{22}(s)\Delta(s)\right)^{-1} G_{21}(s), \\ G_{11} &= G_o = (A, B, C, D), G_{12} = \left(A, E_\eta, C, F_\eta\right), \\ G_{21} &= \left(A, B, C_\gamma, D_\gamma\right), G_{22} = \left(A, E_\eta, C_\gamma, F_\gamma\right). \end{aligned}$$

Note that the transfer function G_Δ can be re-written as

$$\begin{aligned} G_\Delta &= \hat{M}_o^{-1}\left(\hat{N}_o + \hat{N}_{12}\Delta\left(I - G_{22}\Delta\right)^{-1} G_{21}\right), \\ \hat{N}_{12} &= \left(A - L_o C, E_\eta - L_o F_\eta, C, F_\eta\right), \end{aligned}$$

where L_o is the observer gain adopted in the normalised LCF $G_o = \hat{M}_o^{-1}\hat{N}_o$. As a result,

$$\mathcal{K}_\Delta - \mathcal{K}_o = \begin{bmatrix} -\Delta_{\hat{N}} & 0 \end{bmatrix}, \Delta_{\hat{N}} = \hat{N}_{12}\Delta\left(I - G_{22}\Delta\right)^{-1} G_{21}.$$

Hence, in the closed-loop configuration, the dynamics of the residual generator (9.17) is governed by

$$r = -R\left(I + \Delta_{\hat{N}}U\right)^{-1}\Delta_{\hat{N}}M_o\hat{V}v.$$

Recall

$$\mathcal{K}_\Delta^{cl} - \mathcal{K}_o^{cl} = \left(\mathcal{K}_\Delta - \mathcal{K}_o\right)\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} = -\Delta_{\hat{N}}\begin{bmatrix} M_o & -U \end{bmatrix}.$$

Thus, for all $\mathcal{K}_\Delta^{cl}$ satisfying,

$$\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_\Delta^{cl}\right) \leq \delta_\Delta^{cl} < 1,$$

it holds,

$$\|r\|_2 \leq \frac{\beta\delta_\Delta^{cl}}{\sqrt{1 - \left(\delta_\Delta^{cl}\right)^2}}\bar{w}, \bar{w} = \left\|\hat{V}v\right\|_2.$$

9.3.4 Fault Detection Performance Indicators

We are now going to apply $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metric as a tool for the introduction of some FD performance indicators, including indicators for fault detectability and fault-to-uncertainty ratio, as a measurement of detectability in uncertain systems.

The objective of introducing a performance indicator is to study, from the system structural point of view, how far a multiplicative fault in form of a left coprime factor can be detected. It is evident from the residual dynamics that, if $\left(\hat{M}_f, -\hat{N}_f\right)$ is close to $\left(\hat{M}_o, -\hat{N}_o\right)$, the fault detectability will become weak. Hence, it is reasonable to apply $\mathcal{K}$-gap as well as $\mathcal{L}_2$-gap metrics to quantify the fault detectability.

A fault detectability indicator for closed-loop systems

Remember that $\forall Q \in \mathcal{H}_\infty$, the residual dynamics, in the faulty case, can be expressed as

$$r = R\left[-\hat{N}_o \ \hat{M}_o\right]\begin{bmatrix} u \\ y \end{bmatrix} = R\left(\left[-\hat{N}_o \ \hat{M}_o\right] - Q\left[-\hat{N}_f \ \hat{M}_f\right]\right)\begin{bmatrix} u \\ y \end{bmatrix}.$$

Notice further

$$\begin{aligned}
\begin{bmatrix} u \\ y \end{bmatrix} &= \begin{bmatrix} I & -K \\ -G_f & I \end{bmatrix}^{-1}\begin{bmatrix} I \\ 0 \end{bmatrix} v = \begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N}_f & \hat{M}_f \end{bmatrix}^{-1}\begin{bmatrix} \hat{V} \\ 0 \end{bmatrix} v \\
&= \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}\left(I + \begin{bmatrix} 0 & 0 \\ -\Delta_{\hat{N}_f} & \Delta_{\hat{M}_f} \end{bmatrix}\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}\right)^{-1}\begin{bmatrix} \hat{V} \\ 0 \end{bmatrix} v \\
&= \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}\begin{bmatrix} I \\ -\left(I + \left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1}\left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} M_o \\ N_o \end{bmatrix} \end{bmatrix}\hat{V} v, \\
\mathcal{K}_f - \mathcal{K}_o &= \left[-\Delta_{\hat{N}_f} \ \ \Delta_{\hat{M}_f}\right] = \left[-\hat{N}_f \ \ \hat{M}_f\right] - \left[-\hat{N}_o \ \ \hat{M}_o\right].
\end{aligned}$$

Now, let

$$\begin{aligned}
\mathcal{K}_f^{cl} - \mathcal{K}_o^{cl} &= \left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}, \\
Q^* &= \arg \inf_{Q \in \mathcal{H}_\infty} \left\| \mathcal{K}_o^{cl} - Q\mathcal{K}_f^{cl} \right\|_\infty .
\end{aligned}$$

It turns out

$$r = R\left(\mathcal{K}_o^{cl} - Q^*\mathcal{K}_f^{cl}\right)\begin{bmatrix} I \\ -\left(I + \left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1}\left(\mathcal{K}_f - \mathcal{K}_o\right)\begin{bmatrix} M_o \\ N_o \end{bmatrix} \end{bmatrix}\hat{V} v,$$

which yields, on the assumption

$$\left\| \mathcal{K}_o^{cl} - \mathcal{K}_f^{cl} \right\|_\infty \leq \delta_f^{cl} < 1, \tag{9.72}$$

and by means of Lemma 9.4 and the definition of $\mathcal{K}$-gap of the closed-loop system,

$$\|r\|_2 \leq \beta \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right) \sqrt{1 + \frac{\left(\delta_f^{cl}\right)^2}{1 - \left(\delta_f^{cl}\right)^2}}\, \bar{w} = \frac{\beta \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right)}{\sqrt{1 - \left(\delta_f^{cl}\right)^2}} \bar{w}.$$

As a result, the following theorem is proved.

Theorem 9.6 *Given the feedback control loop as shown in Fig. 9.1 with controller (9.19), the residual generator (9.17) and the SKR of the faulty system* $\mathcal{K}_f = \begin{bmatrix} -\hat{N}_f & \hat{M}_f \end{bmatrix}$ *satisfying (9.72), it holds*

$$\|r\|_2 \leq \frac{\beta \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right)}{\sqrt{1 - \left(\delta_f^{cl}\right)^2}} \bar{w}. \tag{9.73}$$

Motivated by this result, we introduce the following definition.

Definition 9.4 *Let* $\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}$ *be the SKRs of the fault-free and faulty systems in the closed-loop configuration. The* $\mathcal{K}$*-gap,*

$$\mathcal{I}_{\mathcal{K}}^{cl} = \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right) \tag{9.74}$$

is called $\mathcal{K}$*-gap indicator for fault detectability in feedback control systems.*

It is worth emphasising that

$$\begin{gathered}\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right) \leq \left\|\mathcal{K}_o^{cl} - \mathcal{K}_f^{cl}\right\|_\infty \\ \Longrightarrow \frac{\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right)}{\sqrt{1 - \left(\delta_f^{cl}\right)^2}} \leq \frac{\delta_f^{cl}}{\sqrt{1 - \left(\delta_f^{cl}\right)^2}}.\end{gathered}$$

In fact,

$$\delta_f^{cl} = \sup_{\mathcal{K}_\Delta^{cl}} \left\{\left\|\mathcal{K}_o^{cl} - \mathcal{K}_\Delta^{cl}\right\|_\infty\right\} = \sup_{\mathcal{K}_\Delta^{cl}} \left\{\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_\Delta^{cl}\right)\right\}$$

can be interpreted as the maximal value of the $\mathcal{K}$-gap for unstructured uncertainty and $\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right)$ as the $\mathcal{K}$-gap for structured fault. Hence, (9.74) provides us with a good estimation of the $\mathcal{L}_2$-upper bound of the residual signal in the faulty case.

For a given threshold J_{th}, it is interesting to notice that if

$$\frac{\beta \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right)}{\sqrt{1 - \left(\delta_f^{cl}\right)^2}} \bar{w} \leq J_{th}, \tag{9.75}$$

then the fault cannot be detected. In other words, (9.75) is a necessary condition for a (multiplicative) fault to become detectable in a feedback control system. It can be seen that in condition (9.75), $\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl},\mathcal{K}_f^{cl}\right)$ is independent of the fault detection system design and determined by the nominal and faulty system models. This is the practical interpretation of the $\mathcal{K}$-gap as an indicator for fault detectability. It is evident that a large $\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl},\mathcal{K}_f^{cl}\right)$ means a reliable detection of the corresponding fault.

A fault detectability indicator for open-loop systems
Recall that in the faulty case

$$r = R\left(\hat{M}_o N_f - \hat{N}_o M_f\right) M_f^{-1} u.$$

On the assumption $M_f^{-1}u \in \mathcal{H}_2$, it holds

$$\|r\|_2 \le \beta \delta_{\mathcal{L}_2}\left(\mathcal{K}_o,\mathcal{K}_f\right)\|\nu\|_2, \nu = M_f^{-1}u. \tag{9.76}$$

In this context, it becomes clear that the $\mathcal{L}_2$-gap metric between $\mathcal{K}_o$ and $\mathcal{K}_f$ also builds an indicator for the fault detectability. Thus, we introduce the following definition.

Definition 9.5 *Let $\mathcal{K}_o, \mathcal{K}_f$ be the SKRs of the fault-free and faulty models, respectively. The $\mathcal{L}_2$-gap metric $\delta_{\mathcal{L}_2}\left(\mathcal{K}_o,\mathcal{K}_f\right)$,*

$$\delta_{\mathcal{L}_2}\left(\mathcal{K}_o,\mathcal{K}_f\right) = \left\|\begin{bmatrix} -\hat{N}_o & \hat{M}_o \end{bmatrix}\begin{bmatrix} M_f \\ N_f \end{bmatrix}\right\|_\infty = \left\|\begin{bmatrix} -\hat{N}_f & \hat{M}_f \end{bmatrix}\begin{bmatrix} M_o \\ N_o \end{bmatrix}\right\|_\infty,$$

is called $\mathcal{L}_2$-gap metric indicator for fault detectability in open-loop systems and denoted by $\mathcal{I}_{\mathcal{L}_2}^{ol}$.

9.3.5 Fault-to-uncertainty Ratio and Fault Detectability in Uncertain Systems

We now briefly address the issue of quantifying the fault detectability in systems with uncertainties. Recall that we have, in the last sub-section, introduced $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metric indicators for the fault detectability. On the other hand, in order to reduce false alarms caused by model uncertainties, threshold setting becomes necessary. It is evident that in case of stronger model uncertainties a higher threshold should be set, in order to keep the false alarm rate to an acceptable level. This will, in turn, reduce fault detectability. Motivated by this observation, we introduce the following definition.

Definition 9.6 *Given the fault detectability indicators $\mathcal{I}_{\mathcal{K}}^{cl}, \mathcal{I}_{\mathcal{L}_2}^{ol}$ for closed- and open-loops, respectively, and the boundedness of the system uncertainties in closed- and open-loops, δ_Δ^{cl} and $\delta_{\Delta,\mathcal{L}_2}$,*

$$\forall \mathcal{K}_{\Delta}^{cl}, \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{o}^{cl}, \mathcal{K}_{\Delta}^{cl}\right) \leq \delta_{\Delta}^{cl},$$
$$\forall \mathcal{K}_{\Delta}, \delta_{\mathcal{L}_2}\left(\mathcal{K}_{o}, \mathcal{K}_{\Delta}\right) \leq \delta_{\Delta,\mathcal{L}_2},$$

respectively, we call

$$R_{F2U}^{cl} = \frac{\mathcal{I}_{\mathcal{K}}^{cl}}{\delta_{\Delta}^{cl}}, R_{F2U}^{ol} = \frac{\mathcal{I}_{\mathcal{L}_2}^{ol}}{\delta_{\Delta,\mathcal{L}_2}} \tag{9.77}$$

fault-to-uncertainty ratio (F2U) of closed- and open-loops, respectively.

In what follows, we are going to apply the R_{F2U} given in (9.77) as an indicator for quantifying the fault detectability in closed- and open-loop configured systems with uncertainties. Recall that a multiplicative fault $\mathcal{K}_f$ cannot be detected if

$$\frac{\beta \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{o}^{cl}, \mathcal{K}_{f}^{cl}\right)}{\sqrt{1-\left(\delta_{f}^{cl}\right)^{2}}} \bar{w} \leq J_{th}$$

for closed-loops and

$$\beta \delta_{\mathcal{L}_2}\left(\mathcal{K}_{o}, \mathcal{K}_{f}\right) \|\nu\|_2 \leq J_{th}$$

for open-loops. Let

$$\Pi_{cl} = \frac{\beta \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{o}^{cl}, \mathcal{K}_{f}^{cl}\right)}{J_{th,cl}\sqrt{1-\left(\delta_{f}^{cl}\right)^{2}}}, \Pi_{ol} = \frac{\beta \delta_{\mathcal{L}_2}\left(\mathcal{K}_{o}, \mathcal{K}_{f}\right) \|\nu\|_2}{J_{th,ol}},$$

and suppose $J_{th,cl}, J_{th,ol}$ are the upper bounds given in Theorem 9.6 and inequality (9.76) for the residual in fault-free operations. It turns out, for closed-loops,

$$\Pi_{cl} = \frac{\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{o}^{cl}, \mathcal{K}_{f}^{cl}\right)}{\delta_{\Delta}^{cl}} \sqrt{\frac{1-\left(\delta_{\Delta}^{cl}\right)^{2}}{1-\left(\delta_{f}^{cl}\right)^{2}}} = R_{F2U}^{cl} \sqrt{\frac{1-\left(\delta_{\Delta}^{cl}\right)^{2}}{1-\left(\delta_{f}^{cl}\right)^{2}}}. \tag{9.78}$$

For open-loops, on account of

$$M_{\Delta}^{-1} u = M_{\Delta}^{-1} M_f M_f^{-1} u = \left(I + M_o^{-1} \Delta_M\right)^{-1} \left(I + M_o^{-1} \Delta_{M_f}\right) M_f^{-1} u,$$

and moreover on the assumption that

$$\forall \Delta_M, \left\|M_o^{-1} \Delta_M\right\|_{\infty} \leq \delta_M < 1, \left\|M_o^{-1} \Delta_{M_f}\right\|_{\infty} \leq \delta_{M_f},$$

it turns out

$$\begin{aligned} \left\| M_{\Delta}^{-1} u \right\|_2 \leq \frac{1+\delta_{M_f}}{1-\delta_M} \left\| M_f^{-1} u \right\|_2 \Longrightarrow \\ \Pi_{ol} = \frac{\delta_{\mathcal{L}_2}\left(\mathcal{K}_o, \mathcal{K}_f\right)\left(1+\delta_{M_f}\right)}{\delta_{\Delta,\mathcal{L}_2}\left(1-\delta_M\right)} = R_{F2U}^{ol} \frac{1+\delta_{M_f}}{1-\delta_M}. \end{aligned} \tag{9.79}$$

It follows from (9.78) and (9.79) that R_{F2U}^{cl} and R_{F2U}^{ol} are key indicator for the detectability of multiplicative faults in systems with uncertainties. It is important to point out that R_{F2U}^{cl} and R_{F2U}^{ol} are structural property of a dynamic system, and are independent of the observer design. Moreover, although a larger R_{F2U} means a better fault detectability, the fault detectability also depends on other parameters and variables. To be specific,

- it is a function of the uncertainties as well as the input and output signals in the open-loop configured systems,
- in the feedback control system configuration, it depends on the controller parameters.

Example 9.3 *We extend the system model considered in Example 9.2 to include the fault as follows*

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + E\eta(t) + E_f f(t), \\ y(t) &= Cx(t) + Du(t) + F\eta(t) + F_f f(t), \\ \begin{bmatrix} \gamma(t) \\ \varsigma(t) \end{bmatrix} &= \begin{bmatrix} C_\gamma \\ C_\varsigma \end{bmatrix} x(t) + \begin{bmatrix} D_\gamma \\ D_\varsigma \end{bmatrix} u(t) + \begin{bmatrix} F_\gamma \eta(t) \\ F_\varsigma f(t) \end{bmatrix}, \\ \begin{bmatrix} \eta(s) \\ f(s) \end{bmatrix} &= \begin{bmatrix} \Delta(s) & 0 \\ 0 & \Delta_f(s) \end{bmatrix} \begin{bmatrix} \gamma(s) \\ \varsigma(s) \end{bmatrix}, \Delta, \Delta_f \in \mathcal{RH}_\infty, \end{aligned}$$

where Δ, Δ_f represent the uncertainty and fault, respectively, and all system matrices are known and of appropriate dimensions. The transfer function from u to y is described by

$$G_{yu}(s) = G_{11}(s) + G_{\Delta}(s) + G_f(s)$$

with G_{11}, G_{Δ} as given in Example 9.2 and

$$\begin{aligned} G_f &= G_{12,f} \Delta_f \left(I - G_{22,f} \Delta_f\right)^{-1} G_{21,f}, \\ G_{12,f} &= \left(A, E_f, C, F_f\right), G_{21,f} = \left(A, B, C_\varsigma, D_\varsigma\right), \\ G_{22,f} &= \left(A, E_f, C_\varsigma, F_\varsigma\right). \end{aligned}$$

Here, for the sake of simplicity, it is assumed that

$$C_\gamma \left(sI - A\right)^{-1} E_f = 0, C_\varsigma \left(sI - A\right)^{-1} E = 0.$$

An involved but straightforward computation yields

$$\begin{aligned}
\sup_{\mathcal{K}_\Delta^{cl}} \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_\Delta^{cl}\right) &= \sup_{\Delta_{\hat{N}}} \left\| \begin{bmatrix} -\Delta_{\hat{N}} M_o & \Delta_{\hat{N}} U \end{bmatrix} \right\|_\infty, \\
\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right) &= \inf_{Q \in \mathcal{H}_\infty} \left\| \begin{bmatrix} 0 & I \end{bmatrix} - Q \begin{bmatrix} -\Delta_{\hat{N},f} M_o & I + \Delta_{\hat{N},f} U \end{bmatrix} \right\|_\infty, \\
\Delta_{\hat{N},f} &= \hat{N}_{12,f} \Delta_f \left(I - G_{22,f} \Delta_f\right)^{-1} G_{21,f}, \\
\hat{N}_{12,f} &= \left(A - L_o C, E_f - L_o F_f, C, F_f\right)
\end{aligned}$$

with $\Delta_{\hat{N}}$, L_o as defined in Example 9.2, which allows us to calculate

$$R_{F2U}^{cl} = \frac{\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_f^{cl}\right)}{\sup_{\mathcal{K}_\Delta^{cl}} \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_o^{cl}, \mathcal{K}_\Delta^{cl}\right)}.$$

9.4 Fault Isolability

In this and next sections, we address fault isolation issues. In the model-based FDI framework, isolation of additive faults is a mainstream topic which has received considerable research attention. For an LTI system with additive faults, fault isolability is often formulated, due to the system linearity, as a structural property that is independent of the magnitude (size) of the faults under consideration. In this context, a fault isolation is typically achieved by means of a bank of residual generators, the associated residual evaluation and decision logic. The basic procedure consists of (i) clustering of the faults to be isolated, (ii) design of a bank of residual generators in such a way that each of them is (highly) sensitive to a group of defined faults and simultaneously (highly) robust against the other (groups of) faults, and (iii) threshold settings corresponding to the residual banks.

It is evident that different definitions for fault isolability and design schemes are needed when dealing with multiplicative faults. Motivated by our previous study on the application of gap metrics to fault detection issues, whose core is the similarity or distance measurement of two dynamic systems using $\mathcal{K}$-gap metric, we are going to investigate fault isolability and isolation issues with the aid of the gap metric technique. For our purpose, we introduce, analogue to the definition of gap metric given in (9.52), $\mathcal{K}$-gap metric defined by

$$\delta_{\mathcal{K}}\left(\mathcal{K}_1, \mathcal{K}_2\right) = \max\left\{\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1, \mathcal{K}_2\right), \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_2, \mathcal{K}_1\right)\right\}. \tag{9.80}$$

Also, it can be, analogue to (9.53), proved that for $\delta_{\mathcal{K}}\left(\mathcal{K}_1, \mathcal{K}_2\right) < 1$

$$\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1, \mathcal{K}_2\right) = \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_2, \mathcal{K}_1\right) = \delta_{\mathcal{K}}\left(\mathcal{K}_1, \mathcal{K}_2\right). \tag{9.81}$$

Without loss of generality, $\mathcal{K}$-gap metric defined in (9.80) will be applied in the sequel for the evaluation of the similarity or distance of two SKRs.

9.4.1 A Motivation Example

To motivate our study, we consider the nominal system

$$G(s) = \frac{4(s+0.5)}{(s+1)(s+3)},$$

and the faulty plant

$$G_f(s) = \frac{4(s+0.5)}{(s+\varsigma)(s+3)}$$

with ς being a varying parameter reflecting different faults. Let

$$G_{f_1}(s) = \frac{4(s+0.5)}{(s+4)(s+3)}, G_{f_2}(s) = \frac{4(s+0.5)}{(s+0.3)(s+3)}$$

be the two (multiplicative) faulty plant models. It can be computed that

$$\delta_{\mathcal{K}}\left(\mathcal{K}, \mathcal{K}_{f_1}\right) = 0.4878, \delta_{\mathcal{K}}\left(\mathcal{K}, \mathcal{K}_{f_2}\right) = 0.5317.$$

Here, $\mathcal{K}, \mathcal{K}_{f_1}, \mathcal{K}_{f_2}$ represent the SKRs of the nominal and both faulty plant models, respectively. Although the $\mathcal{K}$-gap metric values from the faulty plants 1 and 2 to the nominal plant are similar, the $\mathcal{K}$-gap metric value from the faulty plant 1 to the faulty plant 2,

$$\delta_{\mathcal{K}}\left(\mathcal{K}_{f_1}, \mathcal{K}_{f_2}\right) = 0.8322,$$

is significantly larger, which indicates that faulty plant models $\mathcal{K}_{f_1}$ and $\mathcal{K}_{f_2}$ are quite different.

This example reveals that the distance from the faulty plant model to the nominal plant model cannot sufficiently characterise a multiplicative fault, although it can be successfully applied for fault detection, as demonstrated in the past section. For the purpose of fault isolation, the distance between the faults (models) decides whether a fault can be well isolated from the other faults.

On the other hand, a faulty plant cannot be exactly modelled by a transfer function or an SKR, since a faulty operation is triggered by, for instance, abnormal operation conditions or parameter changes, which are in general a random process. This requires the modelling of faulty operations or plants by means of model clustering. To demonstrate it, let us continue our example and consider

$$G_{f_3}(s) = \frac{4(s+0.5)}{(s+4.5)(s+3)}, G_{f_4}(s) = \frac{4(s+0.5)}{(s+0.35)(s+3)}.$$

Obviously, $G_{f_3}(s)$ and $G_{f_4}(s)$ are slight changes from $G_{f_1}(s)$ and $G_{f_2}(s)$, respectively. This can be confirmed by calculating $\delta_{\mathcal{K}}\left(\mathcal{K}_{f_1}, \mathcal{K}_{f_3}\right), \delta_{\mathcal{K}}\left(\mathcal{K}_{f_2}, \mathcal{K}_{f_4}\right)$,

$$\delta_{\mathcal{K}}\left(\mathcal{K}_{f_1}, \mathcal{K}_{f_3}\right) = 0.0429, \delta_{\mathcal{K}}\left(\mathcal{K}_{f_2}, \mathcal{K}_{f_4}\right) = 0.0615.$$

It is reasonable to cluster $\mathcal{K}_{f_1}$ and $\mathcal{K}_{f_3}$ as well as $\mathcal{K}_{f_2}$ and $\mathcal{K}_{f_4}$ to the same set. It is remarkable to notice that

$$\delta_{\mathcal{K}}\left(\mathcal{K}_{f_2}, \mathcal{K}_{f_3}\right) = 0.8429, \delta_{\mathcal{K}}\left(\mathcal{K}_{f_1}, \mathcal{K}_{f_4}\right) = 0.7971,$$

which implies that $\mathcal{K}_{f_2}$ and $\mathcal{K}_{f_3}$ as well as $\mathcal{K}_{f_1}$ and $\mathcal{K}_{f_4}$ do not belong to the same cluster.

9.4.2 Isolability of Multiplicative Faults

Consider $G_{f_i}(s), i = 1, \cdots, M$, which represent M faulty system operation patterns. The corresponding SKRs are denoted by $\mathcal{K}_{f_i}, i = 1, \cdots, M$.

Definition 9.7 *The set defined by*

$$\mathcal{C}_{f_i} \subseteq \left\{\mathcal{K} : \delta_{\mathcal{K}}\left(\mathcal{K}, \mathcal{K}_{f_i}\right) \leq \delta_i\right\}, 0 < \delta_i < 1, \tag{9.82}$$

is called $\mathcal{C}_{f_i}$ cluster with the cluster center $\mathcal{K}_{f_i}$ and cluster radius δ_i.

It is evident that cluster radius is an indicator for the similarity degree of an element in the cluster to the cluster center. The smaller δ_i is, the higher the similarity degree of the set members to $\mathcal{K}_{f_i}$ becomes.

Definition 9.8 *The faults $\mathcal{K}_{f_i}, i = 1, \cdots, M$, are said to be isolable, if for $i = 1, \cdots, M$,*

$$\forall \mathcal{K} \in \mathcal{C}_{f_i}, \mathcal{K} \notin \mathcal{C}_{f_j}, j \neq i, j = 1, \cdots, M. \tag{9.83}$$

This definition tells us, the faults under consideration are isolable if there exists no overlapping among their corresponding clusters.

Remark 9.6 *In the above definition, it is assumed that there exists no simultaneous existence of two or more faults. Different from the additive faults whose influence on the system dynamics is a linear mapping, the influence of two multiplicative faults cannot be, in general, handled as the sum of the influence of each of these two faults. In other words, if two multiplicative faults occur simultaneously, they should be dealt with together as a faulty operation pattern. That is, they are modelled as a single* *fault. It should be emphasised that this handling does not lead to loss of generality. For example, suppose that we have two (multiplicative) faults $\mathcal{K}_{f_1}, \mathcal{K}_{f_2}$. In case that both of them may occur in the system simultaneously, we define three faulty operation patterns as*

$$\mathcal{K}_{f_1}, \mathcal{K}_{f_2} \text{ and } \mathcal{K}_{f_{1,2}}, \tag{9.84}$$

where $\mathcal{K}_{f_{1,2}}$ is the system SKR, when both faults are present in the system. Logically, according to Definition 9.8, the fault isolation problem of this case is formulated as isolating the three faulty patterns defined in (9.84). In this context, the isolability definition given above is also applicable to the systems with simultaneous faults.

In Sect. 16.3, modelling issues of fault patterns in the probabilistic framework will be studied in detail.

Example 9.4 *In this example, we illustrate the need to define simultaneous faults as a fault pattern. Consider a nominal system with the following state space representation*

$$A = \begin{bmatrix} -1 & 0.8 \\ 0.2 & -0.3 \end{bmatrix}, B = \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.4 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = 0$$

with the corresponding SKR $\mathcal{K}$. Suppose that two multiplicative faults, parameter fault $\mathcal{K}_{f_1}$ and actuator fault $\mathcal{K}_{f_2}$, may simultaneously occur in the system. The parameter fault $\mathcal{K}_{f_1}$ causes changes in A and leads to

$$A_{f_1} = \begin{bmatrix} -1 & 0.2 \\ 0.2 & -0.3 \end{bmatrix},$$

while the actuator fault leads to the change in B as

$$B_{f_2} = \begin{bmatrix} 0.3 & 0.1 \\ 0.8 & 0.4 \end{bmatrix}.$$

It follows from the calculation scheme of $\mathcal{K}$-gap that

$$\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}, \mathcal{K}_{f_1} \right) = 0.4332, \boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}, \mathcal{K}_{f_2} \right) = 0.5251.$$

The $\mathcal{K}$-gap between these two types of faults is

$$\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}_{f_1}, \mathcal{K}_{f_2} \right) = 0.5791.$$

Consider $\mathcal{K}_{f_{1,2}}$ as the system SKR, when both faults are present in the system. Note that

$$\boldsymbol{\delta}_{\mathcal{K}} \left(\mathcal{K}, \mathcal{K}_{f_{1,2}} \right) = 0.5320,$$

and moreover,

$$\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{f_{1,2}},\mathcal{K}_{f_1}\right)=0.5319,\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{f_{1,2}},\mathcal{K}_{f_2}\right)=0.4266.$$

It is clear that the SKR of the faulty system with simultaneous faults $\mathcal{K}_{f_{1,2}}$ is significantly different from the SKR of the system with fault $\mathcal{K}_{f_1}$ and fault $\mathcal{K}_{f_2}$, respectively. In other words, $\mathcal{K}_{f_{1,2}}$ defines a new type of (faulty) system dynamics. Thus, when both faults are present in the system, they should be handled as one fault pattern.

Theorem 9.7 *Given faults $\mathcal{K}_{f_i}$ and the corresponding cluster $\mathcal{C}_{f_i}$ with the cluster center $\mathcal{K}_{f_i}$ and cluster radius δ_i, $i=1,\cdots,M$. They are isolable if*

$$\forall i,j,j\neq i,i,j=1,\cdots,M,\delta_{\mathcal{K}}\left(\mathcal{K}_{f_i},\mathcal{K}_{f_j}\right)>\delta_i+\delta_j. \tag{9.85}$$

Proof Given any $\mathcal{K}\in\mathcal{C}_{f_i}, i\in\{1,\cdots,M\}$, it holds

$$\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_i}\right)\leq\delta_i.$$

For all $j\in\{1,\cdots,M\}, j\neq i$, we have

$$\delta_{\mathcal{K}}\left(\mathcal{K}_{f_i},\mathcal{K}_{f_j}\right)\leq\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_i}\right)+\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_j}\right),$$

which leads to

$$\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_j}\right)\geq\delta_{\mathcal{K}}\left(\mathcal{K}_{f_i},\mathcal{K}_{f_j}\right)-\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_i}\right).$$

Due to (9.85) it yields

$$\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_j}\right)>\delta_i+\delta_j-\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_i}\right)>\delta_j.$$

As a result,

$$\mathcal{K}\notin\mathcal{C}_{f_j},$$

and it follows from Definition 9.8 that the faults under consideration are isolable.

The condition (9.85) is very essential in our study on isolation of multiplicative faults and will be adopted in the sequel as the isolability condition. Note that (9.85) is indeed a sufficient condition for the fault isolability, since the cluster $\mathcal{C}_{f_i}$ is in general a sub-set of

$$\left\{\mathcal{K}:\delta_{\mathcal{K}}\left(\mathcal{K},\mathcal{K}_{f_i}\right)\leq\delta_i\right\},i\in\{1,\cdots,M\}.$$

Suppose that

$$\forall i\in\{1,\cdots,M\},\delta_i=\delta.$$

Then, we have the following corollary.

Corollary 9.2 *Given faults $\mathcal{K}_{f_i}$ and the corresponding cluster $\mathcal{C}_{f_i}$ with the cluster center $\mathcal{K}_{f_i}$ and cluster radius δ, $i=1,\cdots,M$. The faults are isolable if*

$$\min_{\substack{i,j\in\{1,\cdots,M\}\\ i\neq j}} \delta_{\mathcal{K}}\left(\mathcal{K}_{f_i}, \mathcal{K}_{f_j}\right) > 2\delta. \tag{9.86}$$

The proof is straightforward and thus omitted here.

When

$$\forall i, j, j \neq i, i, j = 1, \cdots, M, \delta_{\mathcal{K}}\left(\mathcal{K}_{f_i}, \mathcal{K}_{f_j}\right) = 1, \tag{9.87}$$

the ideal case of fault isolation is achievable. From the mathematical point of view, condition (9.87) means, all the subspaces $\mathcal{K}_{f_i}$, as defined in (9.60), should be orthogonal to each other. This is a very strict condition.

It should be emphasised that the fault isolability definition given in Definition 9.8 and the associated conditions (9.85) and (9.86) describe specified system structural properties, which are independent of fault isolation schemes possibly adopted for the fault isolation purpose.

9.4.3 Formulation of Fault Isolation Problems

With the introduction of the fault isolability definition, we are now in a position to formulate some fault isolation problems.

Optimal system design and configuration aiming at enhancing fault isolability In many applications, fault detection and isolation play a fundamental role to guarantee a reliable and stable system operation. Fault isolability is a system structural property which cannot be changed after the system is constructed. For this reason, fault isolability should be taken into account during the system design and configuration. Theorem 9.7 provides us with a reasonable criterion for an optimal system design and configuration. For instance, sensor allocation is a challenging issue in system design and configuration. For an LTI system, sensor allocation can be formulated as determination of the system output matrix C, which has considerable influence on the fault isolability. Let J be some cost function for the optimal sensor allocation (e.g., for the required control performance). The demands for the fault isolability could be formulated as a constraint and integrated into the following optimisation problem:

$$\begin{gathered}\min_{C} J \\ \text{s.t. } \forall i, j, j \neq i, i, j = 1, \cdots, M, \delta_{\mathcal{K}}\left(\mathcal{K}_{f_i}, \mathcal{K}_{f_j}\right) > \delta_{ij},\end{gathered}$$

where $\mathcal{K}_{f_i}, \mathcal{K}_{f_j}, j \neq i, i, j = 1, \cdots, M$, are SKRs of the fault patterns that are functions of C, and $\delta_{ij} > 0, j \neq i, i, j = 1, \cdots, M$, are some pre-defined constants (indicating the similarity degrees between the faults under consideration) for the fault isolability.

We would like to remark that this topic is of considerable research and practical interests, although it will not be addressed in this book.

Observer- and SKR-based fault isolation Once a fault is detected, fault isolation can be dealt with

- by formulating the isolation problem as a number of fault detection problems, which can then be solved using a bank of observers and observer-based decision units or
- by identifying the fault.

The core of the first fault isolation scheme is the design of the observers and the associated decision units, while the second scheme consists of an identification of the SKR of the faulty system. Both of these two schemes will be addressed in the next section.

9.5 Fault Isolation Schemes

In this section, we consider M faulty system operation patterns represented by clusters $\mathcal{C}_{f_i}, i = 1, \cdots, M$, with $\mathcal{K}_{f_i}$ as the cluster center and δ_i the cluster radius, where $\mathcal{K}_{f_i}$ is the SKR of transfer function matrix $G_{f_i}(s)$. On the assumptions that

- the M faults are isolable,
- fault detection has been successfully performed, and
- the faulty system $G_{f_i}(s)$ is stable,

we will propose various algorithms for achieving fault isolation.

9.5.1 Observer-Based Fault Isolation Algorithms

Analogue to the standard scheme of isolating additive faults, M residual generators corresponding to $\mathcal{K}_{f_i}, i = 1, \cdots, M$, are first constructed. They are driven by the process input and output signals u, y and deliver M residuals, $r_{f_i}, i = 1, \cdots, M$. It is evident that if the system operation is in the $\mathcal{K}_{f_i}$ faulty pattern, r_{f_i} will be significantly weaker than $r_{f_j}, j \neq i, j = 1, \cdots, M$. Based on this principle, residual evaluation with evaluation functions J_i, thresholds $J_{th,i}, i = 1, \cdots, M$, and isolation logic,

$$\begin{cases} J_i \leq J_{th,i}, \\ J_j > J_{th,j}, j \neq i, j = 1, \cdots, M, \end{cases} \Longrightarrow \text{fault in cluster } \mathcal{C}_{f_i}, \qquad (9.88)$$

are then designed towards a fault isolation. This procedure is sketched in Fig. 9.2. Note that for the threshold computation u or v (reference signal) are used corresponding to the open- or closed-loop configuration, respectively.

Next, we discuss about the realisation of this fault isolation scheme in details. We treat the isolation issues for closed- and open-loops separately.

Fault isolation in a closed-loop configuration Given $\mathcal{K}_{f_i}\left(G_{f_i}(s)\right), i = 1, \cdots, M$, our first step is to design M observer-based residual generators. Let

$$\mathcal{K}_{f_i} = \left[-\hat{N}_{f_i}(s) \;\; \hat{M}_{f_i}(s) \right]$$

be the normalised LC (NLC) pair of $G_{f_i}(s)$. As discussed at the beginning of this chapter, we construct a residual generator using the NLC pair as follows

$$r_i(s) = \hat{M}_{f_i}(s)y(s) - \hat{N}_{f_i}(s)u(s), \tag{9.89}$$

which can also be implemented in the state space representation form with the observer gain L_o and post-filter $\bar{\Gamma}$ as given in Theorem 9.1. In the next step, the threshold $J_{th,i}$ will be determined with $\mathcal{L}_2$-norm of r as the evaluation function. Recall that the use of the $\mathcal{K}_{f_i}$-based residual generator serves for the purpose of delivering a strong response to those faults in the fault clusters $\mathcal{C}_{f_j}, j \neq i, j = 1, \cdots, M$, and on the other hand, responding to the faults in the cluster $\mathcal{C}_{f_i}$ weakly. To this end, the threshold $J_{th,i}$ will be set according to

$$J_{th,i} = \sup_{\mathcal{K} \in \mathcal{C}_{f_i}} J_i \tag{9.90}$$

with the cluster radius δ_i. In the sequel, we assume

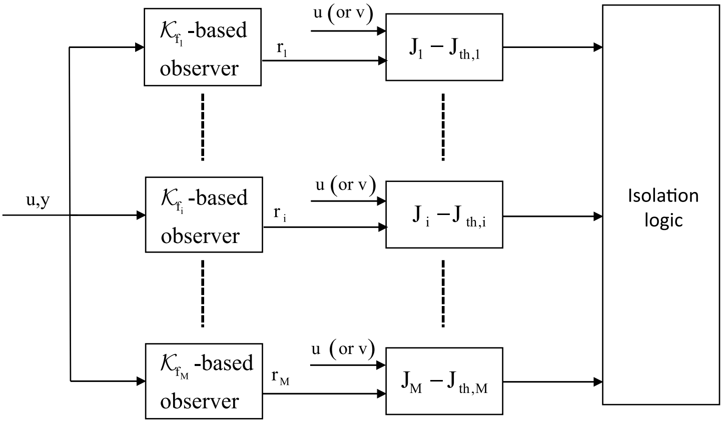

$\mathcal{K}_{f_1}$ -based observer
r_1
u (or v)
$J_1 - J_{th,1}$
u,y
$\mathcal{K}_{f_i}$ -based observer
r_i
u (or v)
$J_i - J_{th,i}$
Isolation logic
$\mathcal{K}_{f_M}$ -based observer
r_M
u (or v)
$J_M - J_{th,M}$

Fig. 9.2 Schematic description of an observer-based fault isolation scheme

$$\mathcal{C}_{f_i} = \left\{\mathcal{K} : \delta_{\mathcal{K}}\left(\mathcal{K}, \mathcal{K}_{f_i}\right) \le \delta_i \in [0, 1)\right\}.$$

In fact, $J_{th,i}$ setting given in (9.90) is equivalent to the threshold setting for systems with model uncertainties, as discussed in Sub-section 9.2.2. To demonstrate it, we first introduce the following lemma.

Lemma 9.6 *Let* $G(s) = \hat{M}^{-1}(s)\hat{N}(s)$ *be the normalised LCF of* G *and*

$$G_1(s) = \left(\hat{M}(s) + \Delta_{\hat{M}}\right)^{-1}\left(\hat{N}(s) + \Delta_{\hat{N}}\right), \Delta_{\hat{N}}, \Delta_{\hat{M}} \in \mathcal{H}_\infty,$$

then for all $0 \le b \le 1$

$$\{G_1 : \delta_{\mathcal{K}}(\mathcal{K}, \mathcal{K}_1) \le b\} = \left\{G_1 : \left\|\begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix}\right\|_\infty \le b\right\}, \tag{9.91}$$

where $\mathcal{K}, \mathcal{K}_1$ *are the SKRs of* $G(s), G_1(s)$, *respectively.*

Lemma 9.6 is the dual result of the well-known relation between the gap metric and the set of right-coprime factor uncertainties, presented in (9.51). It is also a general form of Lemma 9.5. Hence, we omit the proof.

It follows from this lemma that (9.90) can be equivalently written as

$$J_{th,i} = \sup_{\mathcal{C}_{f_i}} J_i = \sup_{\left\{\mathcal{K}:\left\|\mathcal{K}-\mathcal{K}_{f_i}\right\|_\infty \le \delta_i\right\}} J_i, \tag{9.92}$$

in which

$$\Delta\mathcal{K}_{f_i} := \mathcal{K} - \mathcal{K}_{f_i}$$

is treated as uncertainty. Moreover, on the assumption of the system stability, the controller

$$\begin{aligned} K(s) &= -U(s)V^{-1}(s) = -\hat{V}^{-1}(s)\,\hat{U}(s), \\ &\begin{bmatrix} \hat{V} & \hat{U} \end{bmatrix} = \begin{bmatrix} X_o - Q\hat{N}_o & Y_o + Q\hat{M}_o \end{bmatrix}, \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} \hat{Y}_o + M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix}, \end{aligned}$$

stabilises the closed-loop with the plant model $G_{f_i}(s)$. Hence, according to the Youla parameterisation, $K(s)$ can be re-factorised into

$$\begin{aligned} K(s) &= -U_{f_i}(s)V_{f_i}^{-1}(s) = -\hat{V}_{f_i}^{-1}(s)\,\hat{U}_{f_i}(s), \\ \begin{bmatrix} \hat{V}_{f_i} & \hat{U}_{f_i} \end{bmatrix} &= \begin{bmatrix} X_{f_i} - Q_{f_i}\hat{N}_{f_i} & Y_{f_i} + Q_{f_i}\hat{M}_{f_i} \end{bmatrix}, \\ \begin{bmatrix} U_{f_i} \\ V_{f_i} \end{bmatrix} &= \begin{bmatrix} \hat{Y}_{f_i} + M_{f_i}Q_{f_i} \\ \hat{X}_{f_i} - N_{f_i}Q_{f_i} \end{bmatrix}, \\ G_{f_i}(s) &= \hat{M}_{f_i}^{-1}(s)\hat{N}_{f_i}(s) = N_{f_i}(s)M_{f_i}^{-1}(s), \end{aligned} \tag{9.93}$$

where $\left(X_{f_i}, Y_{f_i}\right), \left(\hat{X}_{f_i}, \hat{Y}_{f_i}\right)$ are the transfer matrices given in Bezout identity (9.4) corresponding to the LC and RC pairs of $G_{f_i}(s)$, $\left(\hat{M}_{f_i}, \hat{N}_{f_i}\right)$ and $\left(M_{f_i}, N_{f_i}\right)$, and $Q_{f_i}(s)$ is the parameterisation matrix under the new factorisation.

Remark 9.7 *Note that the factorisation expression (9.93) for controller $K(s)$ holds for all $G_{f_i}(s), i = 1, \cdots, M$.*

Analogue to the discussion in Sub-section 9.2.2, it holds

$$r_i = -\left(I + \Delta\mathcal{K}_{f_i}\begin{bmatrix} -U_{f_i} \\ V_{f_i} \end{bmatrix}\right)^{-1} \Delta\mathcal{K}_{f_i}\begin{bmatrix} M_{f_i} \\ N_{f_i} \end{bmatrix}\hat{V}_{f_i} v.$$

Let

$$\Delta\mathcal{K}_{f_i}^{cl} = \Delta\mathcal{K}_{f_i}\begin{bmatrix} M_{f_i} & -U_{f_i} \\ N_{f_i} & V_{f_i} \end{bmatrix}, \delta_i^{cl} = \sup_{\left\|\Delta\mathcal{K}_{f_i}\right\|_\infty \le \delta_i}\left\{\left\|\Delta\mathcal{K}_{f_i}^{cl}\right\|_\infty\right\}.$$

It follows from the proof of Theorem 9.6 that the threshold is set to be

$$J_{th,i} = \sup_{\mathcal{K}\in\mathcal{C}_{f_i}}\|r_i\|_2 = \frac{\delta_i^{cl}}{\sqrt{1-\left(\delta_i^{cl}\right)^2}}\left\|\hat{V}_{f_i} v\right\|_2. \tag{9.94}$$

Remember that the isolation logic is based on the principle that a fault $\mathcal{K} \in \mathcal{C}_{f_i}$ will result in strong responses in $r_{f_j}, j \neq i, j = 1, \cdots, M$. It is thus of interest to analyse the responses of $J_j, j \neq i, j = 1, \cdots, M$, to such a fault. To simplify our study, we consider $\mathcal{K} = \mathcal{K}_{f_i}$, the center of $\mathcal{C}_{f_i}$. Recalling our study on the fault detection performance in Sub-sections 9.3.4 and 9.3.5, we have

$$J_j = \left\|r_j\right\|_2 \le \frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_{f_i}^{cl}\right)}{\sqrt{1-\left(\delta_{f_{ji}}^{cl}\right)^2}}\left\|\hat{V}_{f_j} v\right\|_2,$$

$$\mathcal{K}_{f_k}^{cl} = \begin{bmatrix} -\hat{N}_{f_k} & \hat{M}_{f_k} \end{bmatrix}\begin{bmatrix} M_{f_j} & -U_{f_j} \\ N_{f_j} & V_{f_j} \end{bmatrix}, k = i, j,$$

$$\left\|\mathcal{K}_{f_j}^{cl} - \mathcal{K}_{f_i}^{cl}\right\|_\infty \le \delta_{f_{ji}}^{cl} < 1.$$

Thus, fault $\mathcal{K}_{f_i}$ leads to

$$\left\|r_j\right\|_2 = J_j > J_{th,j} = \frac{\delta_j^{cl}}{\sqrt{1-\left(\delta_j^{cl}\right)^2}}\left\|\hat{V}_{f_j} v\right\|_2,$$

only if

$$\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl},\mathcal{K}_{f_i}^{cl}\right)}{\delta_j^{cl}} > \sqrt{\frac{1-\left(\delta_{f_{ji}}^{cl}\right)^2}{1-\left(\delta_j^{cl}\right)^2}}. \tag{9.95}$$

Condition (9.95) illustrates that for a given control loop, a successful fault isolation depends considerably on the the distance between the faults, which are expressed by the closed-loop $\mathcal{K}$-gap. Increasing the ratio,

$$\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl},\mathcal{K}_{f_i}^{cl}\right)}{\delta_j^{cl}}, i, j = 1, \cdots, M, j \neq i, \tag{9.96}$$

leads to improvement of the fault isolation performance.

Fault isolation in an open-loop configuration For the residual generation purpose, the same bank of observer-based residual generators like the ones given in (9.89), corresponding to the faulty SKRs $\mathcal{K}_{f_i}\ \left(G_{f_i}(s)\right), i = 1, \cdots, M$, are used. Hence, we begin with our study on the threshold setting, which, similar to the closed-loop configuration, will be done according to (9.90) and using the results achieved in Sub-section 9.2.3.

Consider the residual generator

$$r_i(s) = \hat{M}_{f_i}(s)y(s) - \hat{N}_{f_i}(s)u(s), i = 1, \cdots, M,$$

in the open-loop configuration whose dynamics is governed by

$$\begin{gathered}
r_i = \left(\mathcal{K}_{f_i} - \mathcal{K}\right)\begin{bmatrix} u \\ y \end{bmatrix} = -\Delta\mathcal{K}_{f_i}\begin{bmatrix} I \\ \hat{M}^{-1}\hat{N} \end{bmatrix}u, \\
\begin{bmatrix} -\Delta_{\hat{N}_{f_i}} & \Delta_{\hat{M}_{f_i}} \end{bmatrix} = \begin{bmatrix} -\left(\hat{N} - \hat{N}_{f_i}\right) & \hat{M} - \hat{M}_{f_i} \end{bmatrix} = \Delta\mathcal{K}_{f_i}, \\
\mathcal{K} = \begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} \in \mathcal{C}_{f_i} = \left\{\mathcal{K} : \left\|\mathcal{K} - \mathcal{K}_{f_i}\right\|_\infty \leq \delta_i\right\},
\end{gathered}$$

which yields

$$\begin{aligned}
\|r_i\|_2 &\leq \left\|\Delta\mathcal{K}_{f_i}\right\|_\infty \sqrt{1 + \left\|\hat{M}^{-1}\hat{N}\right\|_\infty^2}\, \|u\|_2 \\
&= \left\|\Delta\mathcal{K}_{f_i}\right\|_\infty \sqrt{1 + \left\|\left(\Delta_{\hat{M}_{f_i}} + \hat{M}_{f_i}\right)^{-1}\left(\Delta_{\hat{N}_{f_i}} + \hat{N}_{f_i}\right)\right\|_\infty^2}\, \|u\|_2 .
\end{aligned}$$

Since

$$\left(\Delta_{\hat{M}_{f_i}} + \hat{M}_{f_i}\right)^{-1}\left(\Delta_{\hat{N}_{f_i}} + \hat{N}_{f_i}\right) = \left(\hat{M}_{f_i}^{-1}\Delta_{\hat{M}_{f_i}} + I\right)^{-1}\hat{M}_{f_i}^{-1}\left(\Delta_{\hat{N}_{f_i}} + \hat{N}_{f_i}\right),$$

on the assumption that

$$\left\| \left[-\hat{M}_{f_i}^{-1} \Delta_{\hat{N}_{f_i}} \ \ \hat{M}_{f_i}^{-1} \Delta_{\hat{M}_{f_i}} \right] \right\|_\infty \leq \delta_{\bar{\Delta}_{f_i}} < 1, \tag{9.97}$$

and using Lemma 9.4, it holds

$$\left\| \left(\Delta_{\hat{M}_{f_i}} + \hat{M}_{f_i} \right)^{-1} \left(\Delta_{\hat{N}_{f_i}} + \hat{N}_{f_i} \right) \right\|_\infty \leq \frac{\delta_{\bar{\Delta}_{f_i}}}{\sqrt{1 - \left(\delta_{\bar{\Delta}_{f_i}} \right)^2}} + \frac{\left\| G_{f_i} \right\|_\infty}{1 - \delta_{\bar{\Delta}_{f_i}}} \Longrightarrow$$

$$\begin{aligned} & \sqrt{1 + \left\| \left(\Delta_{\hat{M}_{f_i}} + \hat{M}_{f_i} \right)^{-1} \left(\Delta_{\hat{N}_{f_i}} + \hat{N}_{f_i} \right) \right\|_\infty^2} \\ & \leq \sqrt{\frac{1}{1 - \left(\delta_{\bar{\Delta}_{f_i}} \right)^2} + \frac{\left\| G_{f_i} \right\|_\infty^2}{\left(1 - \delta_{\bar{\Delta}_{f_i}} \right)^2} + \frac{2\delta_{\bar{\Delta}_{f_i}} \left\| G_{f_i} \right\|_\infty}{\sqrt{1 - \left(\delta_{\bar{\Delta}_{f_i}} \right)^2} \left(1 - \delta_{\bar{\Delta}_{f_i}} \right)}} \\ & \leq \frac{1}{\sqrt{1 - \left(\delta_{\bar{\Delta}_{f_i}} \right)^2}} + \frac{\left\| G_{f_i} \right\|_\infty}{1 - \delta_{\bar{\Delta}_{f_i}}} = \frac{1}{1 - \delta_{\bar{\Delta}_{f_i}}} \left(\sqrt{\frac{1 - \delta_{\bar{\Delta}_{f_i}}}{1 + \delta_{\bar{\Delta}_{f_i}}}} + \left\| G_{f_i} \right\|_\infty \right). \end{aligned}$$

This motivates the threshold setting as

$$J_{th,i} = \sup_{\mathcal{K} \in \mathcal{C}_{f_i}} \| r_i \|_2 = \frac{\delta_i}{1 - \delta_{\bar{\Delta}_{f_i}}} \left(\sqrt{\frac{1 - \delta_{\bar{\Delta}_{f_i}}}{1 + \delta_{\bar{\Delta}_{f_i}}}} + \left\| G_{f_i} \right\|_\infty \right) \| u \|_2 . \tag{9.98}$$

Next, we study the response of J_j, $j \neq i$, $j = 1, \cdots, M$, to fault $\mathcal{K}_{f_i}$ in open-loop configuration. Recall that in the presence of fault $\mathcal{K}_{f_i}$ the dynamics of the residual r_j can be described by

$$r_j = \mathcal{K}_{f_j} \begin{bmatrix} u \\ y \end{bmatrix} = \left(\mathcal{K}_{f_j} - Q \mathcal{K}_{f_i} \right) \begin{bmatrix} u \\ y \end{bmatrix}, Q \in \mathcal{H}_\infty.$$

It turns out

$$
\begin{aligned}
J_j &= \left\| r_j \right\|_2 \leq \delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right) \sqrt{1 + \left\| \hat{M}_{f_i}^{-1} \hat{N}_{f_i} \right\|_\infty^2} \, \|u\|_2 \\
&\leq \frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right)}{1 - \delta_{\bar{\Delta}_{f_{ji}}}} \left(\sqrt{\frac{1 - \delta_{\bar{\Delta}_{f_{ji}}}}{1 + \delta_{\bar{\Delta}_{f_{ji}}}}} + \left\| G_{f_j} \right\|_\infty \right) \|u\|_2 \, , \\
&\text{with } \left\| \left[-\hat{M}_{f_j}^{-1} \Delta_{\hat{N}_{f_{ji}}} \ \ \hat{M}_{f_j}^{-1} \Delta_{\hat{M}_{f_{ji}}} \right] \right\|_\infty \leq \delta_{\bar{\Delta}_{f_{ji}}} < 1, \\
&\mathcal{K}_{f_i} - \mathcal{K}_{f_j} = \left[-\Delta_{\hat{N}_{f_{ji}}} \ \ \Delta_{\hat{M}_{f_{ji}}} \right] = \left[-\left(\hat{N}_{f_i} - \hat{N}_{f_j} \right) \ \hat{M}_{f_i} - \hat{M}_{f_j} \right].
\end{aligned}
$$

As a result, $J_j > J_{th,j}$ only if

$$
\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right)}{\delta_j} > \bar{\Pi}_{o,j}, \tag{9.99}
$$

$$
\bar{\Pi}_{o,j} = \frac{1 - \delta_{\bar{\Delta}_{f_j}}}{1 - \delta_{\bar{\Delta}_{f_{ji}}}} \frac{\sqrt{\frac{1-\delta_{\bar{\Delta}_{f_{ji}}}}{1+\delta_{\bar{\Delta}_{f_{ji}}}}} + \left\| G_{f_j} \right\|_\infty}{\sqrt{\frac{1-\delta_{\bar{\Delta}_{f_j}}}{1+\delta_{\bar{\Delta}_{f_j}}}} + \left\| G_{f_j} \right\|_\infty},
$$

which demonstrates again the ratio

$$
\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right)}{\delta_j}, i, j = 1, \cdots, M, j \neq i,
$$

plays an important role in improving the fault isolation performance.

Fault isolability indicator

Both inequalities (9.95) and (9.99) reveal that the values of $\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_{f_i}^{cl}\right), \delta_j^{cl}$, $\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right), \delta_j, i, j = 1, \cdots, M, j \neq i$, are system structural properties, which determine how far the faults $\mathcal{K}_{f_i}, i = 1, \cdots, M$, can be well isolated using the observer-based isolation schemes. This observation motivates us to introduce the following definition.

Definition 9.9 *Given the SKRs of faults $\mathcal{K}_{f_i}, i = 1, \cdots, M$, the value*

$$
\mathcal{I}_{f_i}^{cl} = \min_{j=1,\cdots,M, j\neq i} \frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_{f_i}^{cl}\right)}{\delta_j^{cl}}, \tag{9.100}
$$

$$
\mathcal{I}_{f_i}^{ol} = \min_{j=1,\cdots,M, j\neq i} \frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right)}{\delta_j}, \tag{9.101}
$$

are called isolability indicators of fault $\mathcal{K}_{f_i}$ in the closed- and open-loop configurations, and the minimum value of $\mathcal{I}_{f_i}, i = 1, \cdots, M$

$$\mathcal{I}_f^{cl} = \min_{i=1,\cdots,M} \mathcal{I}_{f_i}^{cl}, \mathcal{I}_f^{ol} = \min_{i=1,\cdots,M} \mathcal{I}_{f_i}^{ol} \tag{9.102}$$

are called fault isolability indicators in the closed- and open-loop configurations.

It is evident that $\mathcal{I}_{f_i}^{cl}$ ($\mathcal{I}_{f_i}^{ol}$) indicates whether the fault $\mathcal{K}_{f_i}$ could be isolated from the other faults, while $\mathcal{I}_f^{cl}$ ($\mathcal{I}_f^{ol}$) indicates how far all the faults could be isolated from each other. To simplify our study, we assume in the sequel

$$\delta_i^{cl} = \delta^{cl}, \delta_i = \delta^{ol}, i = 1, \cdots, M.$$

It becomes clear that, in order to enhance the fault isolability, the system should be constructed to maximise

$$\min_{j=1,\cdots,M, j\neq i} \delta_{\mathcal{K}} \left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_{f_i}^{cl}\right) \text{ or } \min_{j=1,\cdots,M, j\neq i} \delta_{\mathcal{K}} \left(\mathcal{K}_{f_j}, \mathcal{K}_{f_i}\right)$$

for all $i = 1, \cdots, M$. Furthermore, in terms of the fault isolability indicator, the fault isolability condition given in Corollary 9.2 can also be re-formulated as follows.

Corollary 9.3 *Given faults $\mathcal{K}_{f_i}$ and the corresponding cluster $\mathcal{C}_{f_i}$ with the cluster center $\mathcal{K}_{f_i}$ and cluster radius*

$$\delta = \begin{cases} \delta^{cl}, in\ closed - loop, \\ \delta^{ol}, in\ open - loop, \end{cases}$$

$i = 1, \cdots, M$. They are isolable if

$$\mathcal{I}_f > 2, \mathcal{I}_f = \begin{cases} \mathcal{I}_f^{cl}, in\ closed - loop, \\ \mathcal{I}_f^{ol}, in\ open - loop. \end{cases} \tag{9.103}$$

Proof The proof is evident by re-writing condition (9.86) as

$$\min_{\substack{i,j\in\{1,\cdots,M\} \\ i\neq j}} \delta_{\mathcal{K}} \left(\mathcal{K}_{f_i}^{cl}, \mathcal{K}_{f_j}^{cl}\right) > 2\delta^{cl} \Longleftrightarrow \min_{\substack{i,j\in\{1,\cdots,M\} \\ i\neq j}} \frac{\delta_{\mathcal{K}} \left(\mathcal{K}_{f_i}^{cl}, \mathcal{K}_{f_j}^{cl}\right)}{\delta^{cl}} = \mathcal{I}_f^{cl} > 2,$$

$$\min_{\substack{i,j\in\{1,\cdots,M\} \\ i\neq j}} \delta_{\mathcal{K}} \left(\mathcal{K}_{f_i}, \mathcal{K}_{f_j}\right) > 2\delta^{ol} \Longleftrightarrow \min_{\substack{i,j\in\{1,\cdots,M\} \\ i\neq j}} \frac{\delta_{\mathcal{K}} \left(\mathcal{K}_{f_i}, \mathcal{K}_{f_j}\right)}{\delta^{ol}} = \mathcal{I}_f^{ol} > 2.$$

Finally, we investigate relations between the fault isolability condition (9.103) and the necessary conditions (9.95) and (9.99) for observer-based fault isolation. We first consider inequality (9.95) for a more general case: $\forall \mathcal{K}_i \in \mathcal{C}_{f_i}$. It is straightforward that (9.95) becomes

$$\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_i^{cl}\right)}{\delta_j^{cl}} > \sqrt{\frac{1-\left(\delta_{f_{ji}}^{cl}\right)^2}{1-\left(\delta_j^{cl}\right)^2}},$$
$$\mathcal{K}_i = \begin{bmatrix} -\hat{N}_i & \hat{M}_i \end{bmatrix} \in \mathcal{C}_{f_i}, \mathcal{K}_i^{cl} = \begin{bmatrix} -\hat{N}_i & \hat{M}_i \end{bmatrix} \begin{bmatrix} M_{f_j} & -U_{f_j} \\ N_{f_j} & V_{f_j} \end{bmatrix},$$
$$\left\| \mathcal{K}_{f_j}^{cl} - \mathcal{K}_i^{cl} \right\|_{\infty} \le \delta_{f_{ji}}^{cl}.$$

Recall that

$$\left\| \mathcal{K}_{f_j}^{cl} - \mathcal{K}_i^{cl} \right\|_{\infty} > \delta^{cl}, \delta_j^{cl} = \delta^{cl}, \delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_i^{cl}\right) > \delta^{cl},$$

if the isolation condition (9.103) holds. It is therefore clear that

$$\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}^{cl}, \mathcal{K}_i^{cl}\right)}{\delta_j^{cl}} > 1, \sqrt{\frac{1-\left(\delta_{f_{ji}}^{cl}\right)^2}{1-\left(\delta_j^{cl}\right)^2}} < 1.$$

That means, the necessary condition (9.95) for a successful fault isolation in the closed-loop configuration is satisfied.

In the open-loop configuration, we have, for a more general case $\forall \mathcal{K}_i \in \mathcal{C}_{f_i}$,

$$\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j}, \mathcal{K}_i\right)}{\delta_j} > \frac{1-\delta_{\bar{\Delta}_{f_j}}}{1-\delta_{\bar{\Delta}_{f_{ji}}}} \frac{\sqrt{\frac{1-\delta_{\bar{\Delta}_{f_{ji}}}}{1+\delta_{\bar{\Delta}_{f_{ji}}}}} + \left\| G_{f_j} \right\|_{\infty}}{\sqrt{\frac{1-\delta_{\bar{\Delta}_{f_j}}}{1+\delta_{\bar{\Delta}_{f_j}}}} + \left\| G_{f_j} \right\|_{\infty}},$$
$$\left\| \begin{bmatrix} -\hat{M}_j^{-1} \Delta_{\hat{N}_{f_{ji}}} & \hat{M}_j^{-1} \Delta_{\hat{M}_{f_{ji}}} \end{bmatrix} \right\|_{\infty} \le \delta_{\bar{\Delta}_{f_{ji}}},$$
$$\mathcal{K}_i - \mathcal{K}_{f_j} = \begin{bmatrix} -\Delta_{\hat{N}_{f_{ji}}} & \Delta_{\hat{M}_{f_{ji}}} \end{bmatrix} = \begin{bmatrix} -\left(\hat{N}_i - \hat{N}_{f_j}\right) & \hat{M}_i - \hat{M}_{f_j} \end{bmatrix}.$$

When the isolation condition (9.103) holds, it is reasonable to assume that

$$\delta_{\bar{\Delta}_{f_j}} \le \delta_{\bar{\Delta}_{f_{ji}}},$$

which leads to

$$\frac{\sqrt{\frac{1-\delta_{\bar{\Delta}_{f_{ji}}}}{1+\delta_{\bar{\Delta}_{f_{ji}}}}} + \left\| G_{f_j} \right\|_{\infty}}{\sqrt{\frac{1-\delta_{\bar{\Delta}_{f_j}}}{1+\delta_{\bar{\Delta}_{f_j}}}} + \left\| G_{f_j} \right\|_{\infty}} \le 1.$$

Since, on the assumption of the isolation condition (9.103),

$$\frac{\delta_{\mathcal{K}}\left(\mathcal{K}_{f_j},\mathcal{K}_i\right)}{\delta_j}=\gamma>1$$

it can be concluded that if

$$\frac{1-\delta_{\bar{\Delta}_{f_j}}}{1-\delta_{\bar{\Delta}_{f_{ji}}}}<\gamma,$$

the necessary condition (9.99) for a successful fault isolation in the open-loop configuration is satisfied.

It is worth emphasising that the fault isolability conditions given in Theorem 9.7 and Corollary 9.2 are conditions of the system structure for the fault isolability, while the inequalities given in (9.95) and (9.99) are (necessary) conditions for isolating the faults using the proposed observer-based scheme.

9.5.2 An SKR Identification Based Fault Isolation Strategy

It should be noticed that the observer-based fault isolation algorithms introduced in the previous sub-section do not guarantee a perfect fault isolation, once a fault is detected. Since the conditions given in (9.95) and (9.99) are only necessary so that in the case of fault $\mathcal{K}_{f_i}$

$$\left\|r_j\right\|_2=J_j>J_{th,j},\ j=1,\cdots,M,\ j\neq i,$$

it is possible that

$$\exists j\in\{1,\cdots,M,\ j\neq i\},\ \left\|r_j\right\|_2=J_j\leq J_{th,j}.$$

On the other hand, according to the isolation logic, it holds

$$\|r_i\|_2=J_i\leq J_{th,i}.$$

The consequence is that no decision can be made between fault $\mathcal{K}_{f_i}$ and fault $\mathcal{K}_{f_j}$.

In order to solve the above problem, we propose below an alternative fault isolation scheme, which can be activated when no unique isolation decision could be made, as the above described situation occurs, or directly after a fault is detected. The core of this fault isolation scheme is the (online) identification of the (faulty) SKR and, based on it, a data-driven computation of the $\mathcal{K}$-gap metric.

For our purpose, recall the result in Sect. 4.4 for the data-driven SKR expression (4.76)

$$Y_{k,s}-\hat{Y}_{k,s}=Y_{k,s}-K_pZ_p-K_{f,u}U_{k,s},$$

and denote the SKR by

$$\mathcal{K}_d = \begin{bmatrix} -K_p & -K_{f,u} & I \end{bmatrix},$$

where $\mathcal{K}_d$ is identified using process data sets and their LQ decomposition:

$$\begin{bmatrix} Z_p \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix},$$

$$\begin{bmatrix} K_p & K_{f,u} \end{bmatrix} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{+}.$$

In the next step, a normalisation of $\mathcal{K}_d$, denoted by $\tilde{\mathcal{K}}_d$, is determined by

- first, an SVD of $\mathcal{K}_d$
$$\mathcal{K}_d = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} V^T$$
- then set
$$\tilde{\mathcal{K}}_d = \Sigma^{-1} U^T \mathcal{K}_d. \qquad (9.104)$$

It is evident that $\tilde{\mathcal{K}}_d$ is a data-driven SKR and further

$$\tilde{\mathcal{K}}_d \tilde{\mathcal{K}}_d^T = I.$$

Hence, it is a normalised data-driven SKR.

It is assumed that corresponding to the faulty SKRs, $\mathcal{K}_{f_i}, i = 1, \cdots, M$, the normalised data-driven $\tilde{\mathcal{K}}_{d,f_i}, i = 1, \cdots, M$, have been identified and saved. After a fault is detected, the SKR identification algorithm is activated, which results in $\tilde{\mathcal{K}}_d$. In the next step, computation of $\mathcal{K}$-gap metric based on the data-driven SKRs will be done. To this end, we give the following theorem.

Theorem 9.8 *Let $\tilde{\mathcal{K}}_{d,f_i}$, $\tilde{\mathcal{K}}_d$ be the normalised data-driven SKR for the faulty center and the identified plant. Then, the data-driven realisation of $\mathcal{K}$-gap can be calculated by*

$$\delta_{\mathcal{K}} \left(\mathcal{K}_{d,f_i}, \mathcal{K}_d \right) = \sigma_{\max} \left(\tilde{\mathcal{K}}_d^T - \tilde{\mathcal{K}}_{d,f_i}^T \tilde{\mathcal{K}}_{d,f_i} \tilde{\mathcal{K}}_d^T \right). \qquad (9.105)$$

The computation algorithm given in the above theorem is the dual result on the computation of gap metric based on data-driven SIR, which has been recently reported, as cited at the end of this chapter. Thus, the proof is omitted.

Applying (9.105), the fault isolation is finally achieved by running the following algorithm.

Algorithm 9.1 *Data-driven SIR-based fault isolation*

- *Compute*

$$\delta_{\mathcal{K}} \left(\mathcal{K}_{d,f_i}, \mathcal{K}_d \right) = \sigma_{\max} \left(\tilde{\mathcal{K}}_d^T - \tilde{\mathcal{K}}_{d,f_i}^T \tilde{\mathcal{K}}_{d,f_i} \tilde{\mathcal{K}}_d^T \right), i = 1, \cdots, M; \qquad (9.106)$$

- *Check* $\delta_{\mathcal{K}}\left(\mathcal{K}_{d,f_i},\mathcal{K}_d\right)$ *and make decision*

$$\begin{cases} \delta_{\mathcal{K}}\left(\mathcal{K}_{d,f_i},\mathcal{K}_d\right) > \delta_i \Longrightarrow \mathcal{K}_{f_i} does\ not\ occur, \\ \delta_{\mathcal{K}}\left(\mathcal{K}_{d,f_i},\mathcal{K}_d\right) \leq \delta_i \Longrightarrow \mathcal{K}_{f_i} occurs. \end{cases}$$

Remark 9.8 *In practice,* $\delta_{\mathcal{K}}\left(\mathcal{K}_{d,f_i},\mathcal{K}_d\right)$ *is in general not equal to one. Thus,*

$$\delta_{\mathcal{K}}\left(\mathcal{K}_{d,f_i},\mathcal{K}_d\right) = \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_{d,f_i},\mathcal{K}_d\right) = \boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_d,\mathcal{K}_{d,f_i}\right).$$

It should be kept in mind that an SKR identification requires collection of sufficient process data, and is time and computation consuming. This is a disadvantage of this isolation scheme in comparison with the observer-based scheme proposed in the previous sub-section.

9.6 Notes and References

This chapter is mainly dedicated to the issues of detecting and isolating multiplicative faults in LTI systems with uncertainties. Although our major focus is on a class of multiplicative faults that are modelled in the form of uncertain left coprime factors, the achieved results can also be applied to some other classes of multiplicative faults. The argument for this claim is our discussion in the first section on the equivalent relations between the different types of model uncertainties and faults. Lemma 9.2 adopted in this investigation is given in [1]. The state space computations of the normalised RC and LC pairs, as given in Theorem 9.1, are the well-established results known in robust control, see for instance [2].

Two major issues have been addressed in the first part of this chapter,

- observer-based fault detection system design, including the design of the observer-based residual generator and the threshold setting, and
- system analysis.

Considering that the impact of multiplicative faults on system dynamics considerably depends on the system configuration, we have studied the above two FD issues for open-loop and closed-loop configured systems respectively.

In the closed-loop configuration with a given controller, the optimisation of the observer gain matrix L (for the observer-based residual generator) can be formulated as

$$\min_L \left\| \begin{bmatrix} -U \\ V \end{bmatrix} R^{-1} \right\|_\infty ,$$

where $\begin{bmatrix} -U \\ V \end{bmatrix}$ is the SIR of the controller,

$$R(s) = I - C\,(sI - A + LC)^{-1}\,(L - L_o)\,,$$

with L_o as the observer gain matrix adopted in the computation of the SIR of the controller, which is interpreted as the gain of the observer embedded in the controller. Recall that all stabilisation controllers are residual-driven. The proof of Lemma 9.3 reveals that designing an observer-based residual generator aiming at an optimal fault detection is equivalent to finding an observer-based residual generator that leads to an optimal controller in the sense of the above optimisation problem. It is remarkable that our work reveals a fact that fault detectability and system robustness in the sense of stability margin can be consistently achieved by minimising the H_∞-norm of the SIR of the controller. This issue will be further addressed in our subsequent investigation on fault-tolerant control.

Concerning the threshold setting, the adaptive threshold (9.39) has been derived, which depends on the norm-boundedness of the uncertainties and the controller.

For the observer-based residual generator design of open-loop configurated systems, it has been proved that the normalised SKR of the plant is the optimal residual generator. Based on Lemma 9.4, which is given in [3], the threshold settings (9.44) as well as (9.46) have been derived.

Our study on system analysis is devoted to the analysis of the system structure from the FD aspect. We have introduced the concepts of

- indicators for fault detectability, which is a structural property of the system under consideration and indicates how far a (multiplicative) fault can be detected,
- fault-to-uncertainty ratio (F2U), which is also a system structural property and indicates how far a (multiplicative) fault can be detected in a plant with uncertainties.

For our purpose, the well-established gap metric technique has been applied and extended in our study. The needed preliminaries are introduced in Sub-section 9.3.1, including gap metric, T-gap, ν-gap and $\mathcal{L}_2$-gap metrics. All these results are well described in the book by Vinnicombe [4]. Some of them can also be found in [3, 5, 6]. In our study, the concept of $\mathcal{K}$-gap $\boldsymbol{\delta}_{\mathcal{K}}\left(\mathcal{K}_1, \mathcal{K}_2\right)$ has been introduced for the FD purpose, as a dual form to the directed gap $\boldsymbol{\delta}\left(\mathcal{G}_1, \mathcal{G}_2\right)$. In the proof of Theorem 9.4 in dealing with $\mathcal{K}$-gap computation, the results given in [7] have been applied. The relationships between (unstructured) (left) coprime factor model uncertainties and $\mathcal{K}$-gap metric given in Lemmas 9.5 and 9.6 are the dual results of the relations between (right) coprime factor uncertainties and gap metric proved, for instance, by [3].

Although the $\mathcal{K}$-gap between two SKRs can be computed in the H_∞-norm setting, it is worth paying attention to Definition 9.2, which shows clearly that $\mathcal{K}$-gap is a distance measurement of two kernel subspaces. This definition also holds for other types of systems and allows us, for instance, to realise $\mathcal{K}$-gap computation in the data-driven fashion, as demonstrated in the end of this chapter.

It is the state of the art that multiplicative faults are modelled as parametric faults and detected using the well-established parameter identification technique [8, 9]. In comparison, there are few investigations on the application of observer-based FD methods to detecting multiplicative faults. In fact, it is an open problem to deal with detection and isolation of multiplicative faults in a systematic manner.

It is observed that the analysis of system structure in view of achievable FDI performance is of considerable interests in practical applications. For instance, in their review papers, Wang et al. [10, 11] have demonstrated the importance of fault diagnosability and, associated with it, the system configurability in spacecraft control systems. Although intensive attention has been drawn to the design approaches for FDI systems, limited research efforts have been devoted to the system analysis in the context of FDI performance. In some recent investigations, qualitative FDI performance analysis has been addressed for stochastic systems with additive faults [11–15]. In their recent work, Wang et al. [11] have summarised and analysed the existing methods for the assessment of FDI performance, and described their potential applications in practice. In comparison, few research efforts have been dedicated to the detection and performance analysis issues for multiplicative faults, which can cause considerable changes in the system dynamics, even instability. In particular, few methods are available and applied for the system analysis to give quantitative answers to the questions like how far a multiplicative fault is detectable and how to detect multiplicative faults in systems with influential uncertainties. A quantisation of these features is helpful to get a deep insight into the system structural properties and thus for establishing appropriate design objectives. On the other hand, it is the nature of any model-based framework that model uncertainty issues should be continuously addressed. In order to reduce uncertainty-induced false alarms to an acceptable level, integrating a threshold into an FD system is necessary, which will in turn affect the fault detectability. On account of these observations, it is reasonable to address the threshold setting schemes for uncertain systems and define some FDI performance indicators for multiplicative faults. It is the major intention of our work to apply the gap metric technique to deal with these issues. In summary, we have achieved the following results:

- The $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metric have been introduced to characterise the distance between two kernel subspaces.
- The $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metric aided analysis of residual dynamics with respect to model uncertainties has been presented for both open- and closed-loop configured systems, respectively. Specifically, the concept of $\mathcal{K}$-gap of closed-loop systems has been introduced, which provides a compact and less conservative assessment form of the influence of the model uncertainties on the residual.
- $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metrics have been applied to the performance analysis of fault detection systems from system structure aspect by introducing the indicators for fault detectability and F2U.

The second part of this chapter has been dedicated to the issues of isolating multiplicative faults. To our knowledge, there are few results reported on this topic. In fact, a systematic formulation of isolability and isolation of multiplicative faults is missing. This has motivated us to introduce the definition of the fault cluster characterised by the cluster center and cluster radius. On this basis, fault isolability is defined as a system structural property, and the fault isolation problems are formulated. We hope, by this work, a framework for fault isolation study is established. It is worth remarking that the mathematical tool adopted in building this framework is the $\mathcal{K}$-gap metric.

For the (online) fault isolation purpose, we have proposed two schemes following two different strategies. The first one is to formulate the isolation problem as a number of fault detection problems, which are then solved using a bank of observers and observer-based decision units. To this end, a number of observer-based residual generators corresponding to the (fault) cluster centers are constructed. The threshold setting has adopted the algorithms proposed for the threshold determination aiming at fault detection. Also in this work, the $\mathcal{K}$-gap metric based method has played an important role in the analysis of fault isolation performance.

Different from the observer-based algorithms, the second fault isolation scheme is based on the identification of the (faulty) SKR of the process under consideration once a fault is detected. To this end, the well-established result for the SKR identification, which is reviewed in Sect. 4.4 and is also called data-driven realisation of system SKR, has been adopted. A fault isolation is then achieved by a fault classification, which is formulated and realised on the basis of the definition of the fault cluster with its cluster center and cluster radius. For the needed computation of the $\mathcal{K}$-gap metric between two data-driven SKRs, Theorem 9.8 is introduced. This theorem is the dual result on the computation of gap metric based on data-driven SIR, which has been recently reported in [16].

It should be remarked that these two fault isolation schemes follow different strategies and thus are different in their performance and implementation. While the observer-based isolation scheme is powerful in performing real-time fault isolation but limited in its performance, the SKR identification based one delivers high isolation performance but requires collecting sufficient process data, and is thus time and computation consuming. Both fault isolation schemes could be applied in combination.

At the end of this chapter, we would like to mention that, in combination with other design methods and techniques, the results achieved in this chapter can also be applied to dealing with FDI system design issues, for instance,

- application of the randomised algorithm technique [17] to the $\mathcal{K}$-gap based FDI system design in the probabilistic framework,
- application of the well-established μ-synthesis technique [18] to the $\mathcal{K}$-gap aided FDI system using the LFT models, and
- application of $\mathcal{K}$-gap and $\mathcal{L}_2$-gap metric of finite frequency range [19] to the FDI system design.

References

1. G. Gu and L. Qiu, "Connection of multiplicative/relative perturbation in coprime factors and gap metric uncertainty," *Automatica*, vol. 34, pp. 603–607, 1998.
2. K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, New Jersey: Prentice-Hall, 1996.
3. T. T. Georgiou and M. C. Smith, "Optimal robustness in the gap metric," *IEEE Trans. AC*, vol. 35, pp. 673–686, 1990.
4. G. Vinnicombe, *Uncertainty and Feedback: H_{inf} Loop-Shaping and the V-Gap Metric*. World Scientific, 2000.
5. K. Zhou, *Essential of Robust Control*. Englewood Cliffs, NJ: Prentice-Hall, 1998.
6. A. Feintuch, *Robust Control Theory in Hilbert Space*. New York: Springer-Verlag, 1998.
7. T. T. Georgiou, "On the computation of the gap metric," *Syst. Contr. Letters*, vol. 11, pp. 253–257, 1988.
8. R. Isermann, "Process fault detection based on modeling and estimation methods - a survey," *Automatica*, vol. 20, pp. 387–404, 1984.
9. R. Isermann, *Fault Diagnosis Systems*. Berlin Heidelberg: Springer-Verlag, 2006.
10. D.-Y. Wang, Y.-Y. Tu, C.-R. Liu, Y.-Z. He, and W.-B. Li, "Conotation and research of reconfigurability for space control systems: A review," *Acta Automatica Sinica*, vol. 43, pp. 1687–1702, 2017.
11. D.-Y. Wang, F.-Z. Fu, C.-R. Liu, W.-B. Li, W.-J. Liu, Y.-Z. He, and Y. Xing, "Connotation and research status of diagnosability of control systems: A review," *Acta Automatica Sinica*, vol. 44, pp. 1537–1553, 2018.
12. D. Eriksson, M. Krysander, and E. Frisk, "Quantitative stochastic fault diagnosability analysis," *Proc. the 50th IEEE CDC*, 2011.
13. D. Eriksson, E. Frisk, and M. Krysander, "A method for quantitative fault diagnosability analysis of stochastic linear descriptor models," *Automatica*, vol. 49, pp. 1591–1600, 2013.
14. M. Nyberg, "Criterions for detectability and strong detectability of faults in linear systems," *Int. J. Contr.*, vol. 75, pp. 490–501, 2002.
15. F.-Z. Fu, D.-Y. Wang, and W.-B. Li, "Quantitative evaluation of actual LOE fault diagnosability for dynamic systems," *Acta Automatica Sinca*, vol. 43, pp. 1941–1949, 2017.
16. K. Koenings, M. Krueger, H. Lou, and S. X. Ding, "A data-driven computation method for the gap metric and the optimal stability margin," *IEEE Trans. on Automatic Control*, vol. 63, pp. 805–810, 2018.
17. S. X. Ding, L. Li, and M. Kruger, "Application of randomized algorithms to assessment and design of observer-based fault detection systems," *Automatica*, vol. 107, pp. 175–182, 2019.
18. D. Henry and A. Zolghadri, "Design and analysis of robust residual generators for systems under feedback control," *Automatica*, vol. 41, pp. 251–264, 2005.
19. T. Iwasaki and S. Hara, "Generalized KYP leamma: Unified frequency domain inequalities with design applications," *IEEE Trans. on Autom. Contr.*, vol. 50, pp. 41–59, 2005.

Part III
Fault Detection in Nonlinear Dynamic Systems

Chapter 10
Analysis and Design of Observer-Based Fault Detection Systems

Although there exist many open issues in dealing with fault detection and estimation in linear systems, as can be seen from our works in the previous chapters, the most challenging topic in the fault detection research and application areas is nonlinear observer-based fault detection (FD). A review of the literature in the past decades shows that the application of nonlinear observer theory built the main stream in the nonlinear observer-based FD study in the 90s. In recent years, much attention has been paid to the application of some techniques to addressing nonlinear FD issues, which are newly established for dealing with analysis and synthesis of nonlinear dynamic systems more efficiently. For instance, fuzzy technique based FD, adaptive fault diagnosis for nonlinear systems, LPV (linear parameter varying) based FD or sliding mode observer-based fault detection have been reported.

As we have learnt, an observer-based FD system consists of an observer-based residual generator, a residual evaluator and a decision maker with an embedded threshold. Reviewing the publications on nonlinear observer-based FDI studies reveals that the major research focus in this area is on the design of nonlinear observer-based residual generators. Serving as a major methodology, nonlinear observer theory is widely applied for the investigation. While the early studies have been mainly devoted to the application of feedback-based linearisation, differential algebra and geometric approach techniques to observer-based residual generator design, the current research efforts concentrate on systems with a special class of nonlinearities, typically Lipschitz nonlinearity, sector bounded nonlinearity or special types of control systems like nonlinear switched systems and networked control systems. Differently, some recent works have investigated residual evaluation, threshold setting in the context of performance optimisation of nonlinear observer-based FD systems. In summary, it can be observed that (i) only few of the reported studies have dealt with residual generator and evaluation as well as decision making in an integrated way, and (ii) most of efforts have been made on the FD system design but only few on analysis issues.

Concerning analysis of nonlinear FD systems, it is a surprising observation that little attention has been paid to the existence conditions of nonlinear observer-based FD systems and there is, to our best knowledge, no commonly used conditions for

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_10

checking the existence of an observer-based FD system for general type of nonlinear systems, although this is a fundamental issue for the design of any type of nonlinear observer-based FD systems. On the other hand, rich studies on the input-state, input-output stability and stabilisation of nonlinear systems have been reported in the past decades. And the published results have considerably promoted the development of nonlinear control systems and techniques.

The first objective of this chapter is to investigate existence conditions for a general type of nonlinear observer-based FD systems, which would help us to gain a deeper insight into the fundamental properties of nonlinear observer-based FD systems. This is also the basis for the development of methods for an integrated design of observer-based nonlinear FD systems, the second objective of this chapter.

10.1 Preliminaries and Problem Formulation

10.1.1 System Models

Consider nonlinear systems described by

$$\Sigma : \dot{x} = f(x, u), y = h(x, u), \tag{10.1}$$

where $x \in \mathcal{R}^n, u \in \mathcal{R}^p, y \in \mathcal{R}^m$ denote the state, input and output vectors, respectively. $f(x, u)$ and $h(x, u)$ are continuously differentiable nonlinear functions with appropriate dimensions. The affine form of Σ,

$$\Sigma : \dot{x} = a(x) + B(x)u, y = c(x) + D(x)u \tag{10.2}$$

with $a(x)$, $B(x)$, $c(x)$ and $D(x)$ being continuously differentiable and of appropriate dimensions, is a class of nonlinear systems which are widely adopted in nonlinear system research. This class of nonlinear systems can be considered as a natural extension of LTI systems studied in our previous chapters. Analog to the FD study on LTI systems with additive faults, the fault model of the form

$$\Sigma_w : \begin{cases} \dot{x} = a(x) + B(x)u + E_w(x)w, \\ y = c(x) + D(x)u + F_w(x)w, \end{cases} \tag{10.3}$$

is adopted in our study for modelling nonlinear faulty systems, where $E_w(x)$ and $F_w(x)$ are continuously differentiable nonlinear functions (matrices) that are known and of appropriate dimensions. $w \in \mathcal{R}^{k_w}$ is an unknown vector. $E_w(x)w$ and $F_w(x)w$ represent the influences of the faults on the system dynamics and measurement vector, respectively. The system is called fault-free, when $w = 0$. A more general form of (10.3) is

$$\Sigma_w : \dot{x} = f(x, u, w), y = h(x, u, w). \tag{10.4}$$

10.1.2 Observer-Based Nonlinear Fault Detection Systems

A standard observer-based FD system consists of an observer-based residual generator, a residual evaluator and a decision maker with a threshold. For nonlinear residual generators, we introduce the following definition.

Definition 10.1 *Given the nonlinear system (10.1), a system of the form*

$$\dot{\hat{x}} = \phi(\hat{x}, u, y), \hat{x} \in \mathcal{R}^n, \tag{10.5}$$

$$r = \varphi(\hat{x}, u, y), \tag{10.6}$$

is called observer-based residual generator, if it delivers a residual vector r satisfying that

(i) for $\hat{x}(0) = x(0)$,

$$\forall u, r(t) \equiv 0,$$

(ii) for some $w \neq 0$ *in the faulty system (10.4),* $r(t) \neq 0$.

In order to avoid loss of information about the faults, the residual vector should generally have the same dimension like the output vector. For the sake of simplicity, also considering the conditions (i) and (ii), we suppose that

$$r = \varphi(\hat{x}, u, y) = y - \hat{y}, \hat{y} = h(\hat{x}, u). \tag{10.7}$$

Residual evaluation serves the purpose of making a right decision for a successful fault detection. To this end, a positive definite function of $r(t)$, J, is adopted as residual evaluation function, where positive definite functions will be defined below. We define the threshold as

$$J_{th} = \sup_{x_0, w=0} J, \tag{10.8}$$

which is clearly interpreted as the maximum influence of uncertainties on the residual vector $r(t)$ in the fault-free operation $(w(t) = 0)$. We adopt a simple form of detection logic of the form

$$\text{detection logic: } \begin{cases} J > J_{th} \Longrightarrow \text{faulty}, \\ J \leq J_{th} \Longrightarrow \text{fault-free}. \end{cases} \tag{10.9}$$

10.1.3 Problem Formulation

In the subsequent work in this chapter, two essential nonlinear FD issues will be addressed. The first one deals with the existence conditions of the observer-based

FD system with the residual generator given in Definition 10.1, residual evaluation function, threshold (10.8) and detection logic (10.9). It is worth to emphasise that our work is devoted to the overall observer-based FD system with the residual generator, the evaluator and the decision maker. The second issue is the design of nonlinear observer-based FD systems for affine systems (10.2).

10.1.4 Notation

For our purpose, we introduce some definitions and notations which are known in nonlinear stability theory and will be needed in the subsequent study. Let $\mathcal{R}_+ = [0, \infty)$.

- A function $\gamma : \mathcal{R}_+ \to \mathcal{R}_+$ is said to belong to class $\mathcal{K}$ if it is continuous, strictly increasing, and satisfies $\gamma(0) = 0$. If, in addition, $\lim_{t\to\infty} \gamma(t) = \infty$, then γ belongs to class $\mathcal{K}_\infty$.
- A function $\beta : \mathcal{R}_+ \to \mathcal{R}_+$ is said to belong to class $\mathcal{L}$ if it is continuous, strictly decreasing, and satisfies $\lim_{s\to\infty} \beta(s) = 0$.
- A function $\phi(s, t) : \mathcal{R}_+ \times \mathcal{R}_+ \to \mathcal{R}_+$ is said to belong to class $\mathcal{KL}$ if for each fixed t the function is of class $\mathcal{K}$ and for each fixed s it is of class $\mathcal{L}$.
- Notation $||\cdot||$ stands for the Euclidean norm of a vector in some Euclidean space and
$$\mathcal{B}_r := \{x \in \mathcal{R}^n : ||x|| \le r \text{ for some } r > 0\}.$$
- $\mathcal{L}_2(0, \infty)$ is the space of functions $u : \mathcal{R}_+ \to \mathcal{R}^p$ which are measurable and satisfy
$$\int_0^\infty ||u(t)||^2 dt < \infty.$$
- $\mathcal{L}_{2,[0,\tau]}$-norm of $u(t)$ is defined and denoted by
$$\|u_\tau\|_2 = \left(\int_0^\tau ||u(t)||^2 dt\right)^{1/2},$$
- and $\mathcal{L}_\infty$-norm of $u(t)$ by
$$\|u\|_\infty = ess \sup \{||u(t)||, t \ge 0\}.$$
- A function $f : \mathcal{R}^n \to \mathcal{R}$ is positive definite if $f(x) > 0$ for all $x > 0$, and $f(0) = 0$.
- By $V_{x,\hat{x}}(x, \hat{x})$ we denote
$$V_{x,\hat{x}}(x, \hat{x}) = \left[\, V_x(x, \hat{x}) \;\; V_{\hat{x}}(x, \hat{x}) \,\right] = \left[\, \tfrac{\partial V(x,\hat{x})}{\partial x} \;\; \tfrac{\partial V(x,\hat{x})}{\partial \hat{x}} \,\right].$$

10.2 On Observer-Based FD Systems

In this section, we define two classes of nonlinear observer-based FD systems and study their existence conditions.

10.2.1 *Two Classes of Observer-Based FD Systems*

Given a residual vector, for the residual evaluation purpose, two norm-based evaluation functions are considered in our work:

- the Euclidean norm-based instant evaluation

$$J_E = \alpha_1 \left(\| r \| \right), \tag{10.10}$$

- the integral evaluation with an evaluation window $[0, \tau]$

$$J_2 = \int_0^\tau \alpha_2 \left(\| r \| \right) dt, \tag{10.11}$$

  where $\alpha_1 \left(\| r \| \right), \alpha_2 \left(\| r \| \right)$ are some $\mathcal{K}$-functions.

Definition 10.2 *Given the nonlinear system (10.1), a dynamic system is called*

- *$\mathcal{L}_\infty$ observer-based FD system, when it consists of the observer-based residual generator (10.5) and (10.7), residual evaluation function (10.10) and detection logic (10.9) with a corresponding threshold,*
- *$\mathcal{L}_2$ observer-based FD system, when it consists of the observer-based residual generator (10.5) and (10.7), residual evaluation function (10.11) and detection logic (10.9) with a corresponding threshold.*

In the subsequent two subsections, we are going to study the existence conditions of the above two types of FD systems as well as the construction of the corresponding thresholds.

10.2.2 *On $\mathcal{L}_\infty$ observer-based FD systems*

For our purpose, we first introduce the following definition, which is motivated by the so-called *weak detectability,* known and widely used in the study on the stabilisation of nonlinear systems by output feedback.

Definition 10.3 *System (10.1) is said to be output re-constructible if there exist*

- *a function $\phi : \mathcal{R}^n \times \mathcal{R}^p \times \mathcal{R}^m \to \mathcal{R}^n$,*
- *functions $V(x, \hat{x}) : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}^+$, $\varphi_i(\cdot) \in \mathcal{K}, i = 1, 2, 3$, and*
- *positive constants δ, δ_u,*

such that $\forall x, \hat{x} \in \mathcal{B}_\delta$, $\|u\|_\infty \leq \delta_u$,

$$\varphi_1(\|r\|) \leq V(x, \hat{x}) \leq \varphi_2\left(\left\|x - \hat{x}\right\|\right), r = y - h(\hat{x}, u), \tag{10.12}$$

$$V_x(x, \hat{x}) f(x, u) + V_{\hat{x}}(x, \hat{x}) \phi(\hat{x}, u, y) \leq -\varphi_3\left(\left\|x - \hat{x}\right\|\right). \tag{10.13}$$

Remark 10.1 *Substituting $\|r\|$ in $\varphi_1(\|r\|)$ by $\left\|x - \hat{x}\right\|$, Definition 10.3 becomes equivalent with the well-known weak detectability. If it is further assumed that*

$$\|h(\zeta, u) - h(\varsigma, u)\| \leq \gamma(\|\zeta - \varsigma\|)$$

for some $\gamma \in \mathcal{K}$, then we have

$$\left\|x - \hat{x}\right\| \geq \gamma^{-1}(\|r\|),$$

which leads to

$$\varphi_1\left(\left\|x - \hat{x}\right\|\right) \geq \varphi_1\left(\gamma^{-1}(\|r\|)\right).$$

Since $\varphi_1\left(\gamma^{-1}(\cdot)\right) \in \mathcal{K}$, the weak detectability implies the output re-constructability.

Remark 10.2 *Consider the (overall) system dynamics*

$$\dot{x} = f(x, u), y = h(x, u), \dot{\hat{x}} = \phi(\hat{x}, u, y), r = y - h(\hat{x}, u)$$

with u as its input and r as output. Function $V(x, \hat{x})$ satisfying (10.12)–(10.13) can be understood as a variant of the IOS (input-output stability) Lyapunov function. In fact, the residual generation problem can also be studied in the IOS context. The motivation of introducing the output re-constructability is that in the model-based FDI framework, the output estimate $\hat{y}$ is called analytical redundancy, and residual generation is equivalent with building analytical redundancy.

The following theorem presents a major property of an output re-constructible system, which provides us with a sufficient condition for the existence of an $\mathcal{L}_\infty$ observer-based FD system and the threshold setting.

Theorem 10.1 *Assume that system (10.1) is output re-constructible. Then, system (10.5) with (10.7) as its output delivers a residual vector $r(t)$, and it holds*

$$\|r(t)\| \leq \beta\left(\left\|x(0) - \hat{x}(0)\right\|, t\right), \tag{10.14}$$

where $\beta\left(\left\|x(0) - \hat{x}(0)\right\|, t\right) \in \mathcal{KL}$.

Proof It follows from (10.12) that

$$\left\| x(t) - \hat{x}(t) \right\| \geq \varphi_2^{-1} \left(V(x, \hat{x}) \right),$$

by which (10.13) can be further re-written into

$$\dot{V}(x, \hat{x}) \leq -\varphi_3 \left(\varphi_2^{-1} \left(V(x, \hat{x}) \right) \right).$$

Since $\varphi_3(\varphi_2^{-1}) \in \mathcal{K}$, it is known from the highly cited paper by Sontag in 1989 (see the reference given at the end of this chapter) that there exists a $\mathcal{KL}$-function γ so that

$$V(x(t), \hat{x}(t)) \leq \gamma \left(V(x(0), \hat{x}(0)), t \right).$$

Note that (10.12) yields

$$\|r(t)\| \leq \varphi_1^{-1} \left(V(x, \hat{x}) \right),$$

which, considering (10.12) and (10.13), results in

$$\|r(t)\| \leq \varphi_1^{-1} \left(\gamma \left(\varphi_2 \left(\left\| x(0) - \hat{x}(0) \right\| \right), t \right) \right) =: \beta \left(\left\| x(0) - \hat{x}(0) \right\|, t \right). \tag{10.15}$$

The theorem is thus proved.

Remark 10.3 *A similar proof can be found in the references given at the end of this chapter using an IOS-Lyapunov function, as pointed out in the above remark.*

Remark 10.4 *It is evident from the definition of output re-constructability and the above proof that it also holds*

$$\varphi_1 \left(\|r\| \right) \leq \gamma \left(\varphi_2 \left(\left\| x(0) - \hat{x}(0) \right\| \right), t \right) =: \beta_1 \left(\left\| x(0) - \hat{x}(0) \right\|, t \right), \tag{10.16}$$

where $\beta_1 \left(\left\| x(0) - \hat{x}(0) \right\|, t \right) \in \mathcal{KL}$.

It is obvious that for a given initial estimation error $x(0) - \hat{x}(0)$,

$$\lim_{t\to\infty} \beta \left(\left\| x(0) - \hat{x}(0) \right\|, t \right) = \lim_{t\to\infty} \beta_1 \left(\left\| x(0) - \hat{x}(0) \right\|, t \right) = 0. \tag{10.17}$$

Property (10.17) reveals that the influence of the initial estimation error on the residual evaluation function will disappear with time, as known in the case of LTI systems.

It follows immediately from Theorem 10.1 that the threshold can be schematically set as

$$J_{th} = \beta \left(\delta, 0 \right),$$

if system (10.1) is output re-constructible. Note that J_{th} only depends on the initial estimation error. On the other hand, such a setting could be too conservative. In order to improve the FD performance, the influence of the process input variables on the residual vector should be generally taken into account. It will lead to a so-called adaptive threshold, which allows a more efficient FD. Moreover, considering that the $\mathcal{L}_{2,[0,\tau]}$-norm is often used for the residual evaluation purpose, we are motivated to investigate the following detection scheme.

10.2.3 On $\mathcal{L}_2$ observer-based FD systems

We first introduce the definition of weak output re-constructability.

Definition 10.4 *System (10.1) is said to be weakly output re-constructible if there exist*

- *a function* $\phi : \mathcal{R}^n \times \mathcal{R}^p \times \mathcal{R}^m \to \mathcal{R}^n$,
- *functions* $V(x, \hat{x}) : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}^+$, $\varphi_1(\cdot) \in \mathcal{K}$, $\varphi_2(\cdot) \in \mathcal{K}_\infty$ *and*
- *a constant* $\delta > 0$,

such that $\forall x, \hat{x} \in \mathcal{B}_\delta$

$$V_x(x, \hat{x}) f(x, u) + V_{\hat{x}}(x, \hat{x}) \phi(\hat{x}, u, h(x, u)) \leq -\varphi_1(\|r\|) + \varphi_2(\|u\|). \tag{10.18}$$

Comparing Definitions 10.3 and 10.4, it becomes evident that condition (10.18) is generally weaker than the ones given in Definition 10.3.

The following theorem presents a sufficient condition for the existence of an $\mathcal{L}_2$ observer-based FD system.

Theorem 10.2 *Assume that system (10.1) is weakly output re-constructible. Then, an $\mathcal{L}_2$ observer-based FD system can be realised using functions φ_1, φ_2 and by*

- *constructing residual generator according to (10.5) and (10.7),*
- *defining the evaluation function as*

$$J = \int_0^\tau \varphi_1(\|r(t)\|)\, dt,$$

- *and setting the threshold equal to*

$$J_{th} = \int_0^\tau \varphi_2(\|u\|)\, dt + \bar{\gamma}_o, \bar{\gamma}_o = \sup_{x(0), \hat{x}(0)} \{\gamma_0\}, \gamma_o = V\left(x(0), \hat{x}(0)\right). \tag{10.19}$$

Proof It follows from (10.18) that

$$\dot{V}(x, \hat{x}) \leq -\varphi_1(\|r(t)\|) + \varphi_2(\|u\|).$$

As a result,

$$\int_0^\tau \varphi_1(\|r\|) dt \leq \int_0^\tau \varphi_2(\|u\|) dt + V(x(0), \hat{x}(0)). \tag{10.20}$$

The theorem is thus proved.

In theoretical study on norm-based residual evaluation, it is the state of the art that the evaluation window is assumed to be infinitively large. That is

$$J = \int_0^\infty \varphi_1 \left(\|r\| \right) dt \Longrightarrow J_{th} = \int_0^\infty \varphi_2 \left(\|u\| \right) dt + \bar{\gamma}_o. \tag{10.21}$$

In practice, this is not realistic, since a large evaluation window generally results in a (considerably) delayed fault detection. In dealing with nonlinear FD, a large evaluation window also means a high threshold due to the dependence on u. Considering that a fault may happen after the system is in operation for a long time, for a large evaluation window the influence of w on J may be much weaker than u on J_{th}. As a result, the FD performance can become poor. For these reasons, in practice the evaluation function and threshold are often defined by

$$J = \int_{t_o}^{t_o+\tau} \varphi_1 \left(\|r\| \right) dt \Longrightarrow J_{th} = \int_{t_o}^{t_o+\tau} \varphi_2 \left(\|u\| \right) dt + \bar{\gamma}_o, \tag{10.22}$$

where

$$\bar{\gamma}_o = \sup_{x(t_0), \hat{x}(t_0)} \{\gamma_o\}$$

represents the maximum γ_o for all (bounded) possible $x(t_0), \hat{x}(t_0)$.

In this section, we have derived the existence conditions for two types of nonlinear observer-based FD systems. Although the achieved results do not lead to a direct design of a nonlinear observer-based FD system, they are fundamental for the application of some established nonlinear techniques for FD system design, which is investigated in the subsequent section.

10.3 Design of Observer-Based FD Systems

10.3.1 Design of $\mathcal{L}_\infty$ observer-based FD systems

Suppose that system (10.1) is output re-constructible and for some constant $\delta_o > 0$

$$\left\| x(0) - \hat{x}(0) \right\| \leq \delta_o.$$

Then, we are able to construct residual generator (10.5) using $\phi(\hat{x}, u, y)$ defined in Definition 10.3. Let the evaluation window be $[t_1, t_2]$. It follows from Theorem 10.1 that the residual evaluation function can be defined as

$$J_{E,1} = \varphi_1 \left(\|r(t)\| \right), t \in [t_1, t_2]$$

or alternatively

$$J_E = \|r(t)\|, t \in [t_1, t_2].$$

Corresponding to them, there exist $\mathcal{KL}$-functions $\beta_1\left(\left\|x(0) - \hat{x}(0)\right\|, t\right)$ and $\beta\left(\left\|x(0) - \hat{x}(0)\right\|, t\right)$ satisfying (10.16) and (10.14), respectively. Consider

$$\max_{\substack{\left\|x(0)-\hat{x}(0)\right\| \le \delta_o \\ t\in[t_1,t_2]}} \beta\left(\left\|x(0) - \hat{x}(0)\right\|, t\right)$$
$$= \beta\left(\max_{\left\|x(0)-\hat{x}(0)\right\| \le \delta_o} \left\|x(0) - \hat{x}(0)\right\|, \min_{t\in[t_1,t_2]} t\right) = \beta\left(\delta_o, t_1\right),$$

$$\max_{\substack{\left\|x(0)-\hat{x}(0)\right\| \le \delta_o \\ t\in[t_1,t_2]}} \beta_1\left(\left\|x(0) - \hat{x}(0)\right\|, t\right)$$
$$= \beta_1\left(\max_{\left\|x(0)-\hat{x}(0)\right\| \le \delta_o} \left\|x(0) - \hat{x}(0)\right\|, \min_{t\in[t_1,t_2]} t\right) = \beta_1\left(\delta_o, t_1\right).$$

Finally, the threshold settings are

$$J_{th,1} = \beta_1\left(\delta_o, t_1\right)$$

corresponding to $J_{E,1}$ and

$$J_{th} = \beta\left(\delta_o, t_1\right)$$

corresponding to J_E.

10.3.2 Design of $\mathcal{L}_2$-NFDF for affine systems

We now study the design of $\mathcal{L}_2$ observer-based FD systems for a class of nonlinear systems, the affine systems given in (10.2). We restrict our attention to the following observer-based residual generator

$$\dot{\hat{x}} = a(\hat{x}) + B(\hat{x})u + L(\hat{x})\left(y - c(\hat{x}) - D(\hat{x})u\right), \tag{10.23}$$
$$r = \varphi(\hat{x}, u, y) = y - c(\hat{x}) - D(\hat{x})u, \tag{10.24}$$

which is called nonlinear fault detection filter (NFDF).

Definition 10.5 *Given the nonlinear system (10.2), the NFDF (10.23)–(10.24) is called $\mathcal{L}_2$-NFDF if it satisfies that for some constant $\gamma_u \ge 0$,*

$$\left\|r_\tau\right\|_2^2 \le \gamma_u^2 \left\|u_\tau\right\|_2^2 + \gamma_o, \tag{10.25}$$

where $\gamma_o \geq 0$ is a (finite) constant for given $x(0), \hat{x}(0)$.

It follows from (10.25) that by an $\mathcal{L}_2$-NFDF the threshold can be, on the assumption that γ_o is bounded for all possible $x(0), \hat{x}(0)$, set equal to

$$J_{th} = \gamma_u^2 \left\| u_\tau \right\|_2^2 + \sup_{x(0),\hat{x}(0)} \left\{ \gamma_o \right\}. \tag{10.26}$$

It allows then the application of the following decision logic,

$$\begin{cases} J = \left\| r_\tau \right\|_2^2 > J_{th} \Longrightarrow \text{ faulty,} \\ J = \left\| r_\tau \right\|_2^2 \leq J_{th} \Longrightarrow \text{ fault-free,} \end{cases} \tag{10.27}$$

for a successful fault detection. Note that the $\mathcal{L}_2$-NFDF, the residual evaluation function (10.26) and threshold setting (10.27) build a special realisation of an $\mathcal{L}_2$ observer-based FD system. In fact, we have the following relations

$$J = \left\| r_\tau \right\|_2^2 = \int_0^\tau \left\| r(t) \right\|^2 dt \text{ implies } \varphi_1 \left(\left\| r \right\| \right) = \left\| r \right\|^2,$$
$$J_{th} = \gamma_u^2 \left\| u_\tau \right\|_2^2 + \sup_{x(0),\hat{x}(0)} \left\{ \gamma_o \right\} \text{ implies } \varphi_2 \left(\left\| u \right\| \right) = \left(\gamma_u \left\| u \right\| \right)^2,$$

where φ_1, φ_2 are defined in Theorem 10.2.

Next, we study the design of $\mathcal{L}_2$-NFDF, which means the determination of gain matrix $L(\hat{x})$ so that (10.25) holds. Let

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \bar{f}(x, \hat{x}) + G(x, \hat{x})u + \begin{bmatrix} 0 \\ L(\hat{x})r \end{bmatrix}, \tag{10.28}$$
$$r = y - \hat{y}, \hat{y} = c(\hat{x}) + D(\hat{x})u,$$
$$\bar{f}(x, \hat{x}) = \begin{bmatrix} a(x) \\ a(\hat{x}) \end{bmatrix}, G(x, \hat{x}) = \begin{bmatrix} B(x) \\ B(\hat{x}) \end{bmatrix}.$$

We have the following result.

Theorem 10.3 *Given the system (10.2) and the NFDF (10.23)–(10.24). Suppose that*

- *there exists a constant $\gamma > 0$ so that*

$$\gamma^2 I - \left(D(x) - D(\hat{x}) \right)^T \left(D(x) - D(\hat{x}) \right) > 0, \tag{10.29}$$

and set $\Theta(x, \hat{x})$ given by

$$\gamma^2 I - \left(D(x) - D(\hat{x}) \right)^T \left(D(x) - D(\hat{x}) \right) = \Theta^T(x, \hat{x}) \Theta(x, \hat{x}) \tag{10.30}$$

with $\Theta(x, \hat{x}) = \Theta^T(x, \hat{x})$ being a $p \times p$ matrix,

- *there exists* $V(x,\hat{x}) \geq 0$ *that solves the following Hamilton-Jacobi inequality (HJI)*

$$\begin{gathered}V_{x,\hat{x}}(x,\hat{x})\bar{f}(x,\hat{x}) + \frac{1}{2}\left(c^T(x)c(x) - c^T(\hat{x})c(\hat{x})\right) \\ +\frac{1}{2}w(x,\hat{x})\left(\Theta^T(x,\hat{x})\Theta(x,\hat{x})\right)^{-1}w^T(x,\hat{x}) \leq 0, \\ w(x,\hat{x}) = V_{x,\hat{x}}(x,\hat{x})G(x,\hat{x}) + c^T(x)\left(D(x) - D(\hat{x})\right),\end{gathered} \tag{10.31}$$

- *there exists* $L(\hat{x})$ *solving*

$$V_{\hat{x}}(x,\hat{x})L(\hat{x}) = c^T(\hat{x}). \tag{10.32}$$

Then, it holds

$$\|r_\tau\|_2^2 \leq \gamma^2\|u_\tau\|_2^2 + 2V(x(0),\hat{x}(0)). \tag{10.33}$$

Proof Considering

$$\dot{V}(x,\hat{x}) = V_{x,\hat{x}}(x,\hat{x})\left(\bar{f}(x,\hat{x}) + G(x,\hat{x})u\right) + V_{\hat{x}}(x,\hat{x})L(\hat{x})\left(y - c(\hat{x}) - D(\hat{x})u\right)$$

and (10.32), it holds

$$\dot{V}(x,\hat{x}) = V_{x,\hat{x}}(x,\hat{x})\left(\bar{f}(x,\hat{x}) + G(x,\hat{x})u\right) + \left(\hat{y} - D(\hat{x})u\right)^T\left(y - \hat{y}\right).$$

Note that

$$\begin{aligned}\frac{1}{2}\|r\|^2 &= \frac{1}{2}y^Ty + \frac{1}{2}\hat{y}^T\hat{y} - \hat{y}^Ty, \\ \frac{1}{2}\|y\|^2 &= \frac{1}{2}\|c(x)\|^2 + \frac{1}{2}\|D(x)u\|^2 + c^T(x)D(x)u, \\ \frac{1}{2}\left\|\hat{y}\right\|^2 &= \frac{1}{2}\left\|c(\hat{x})\right\|^2 + \frac{1}{2}\left\|D(\hat{x})u\right\|^2 + c^T(\hat{x})D(\hat{x})u,\end{aligned}$$

and moreover

$$\begin{gathered}\frac{1}{2}\left\|\Theta(x,\hat{x})u - \Theta^{-T}(x,\hat{x})w^T(x,\hat{x})\right\|^2 = \frac{\gamma^2}{2}\|u\|^2 - \frac{1}{2}\left\|\left(D(x) - D(\hat{x})\right)u\right\|^2 \\ -w(x,\hat{x})u + \frac{1}{2}w(x,\hat{x})\left(\Theta^T(x,\hat{x})\Theta(x,\hat{x})\right)^{-1}w^T(x,\hat{x}).\end{gathered}$$

It turns out, by HJI (10.31), that

$$\begin{gathered}\dot{V}(x,\hat{x}) = V_{x,\hat{x}}(x,\hat{x})\left(\bar{f}(x,\hat{x}) + G(x,\hat{x})u\right) - \frac{1}{2}\left(\|r\|^2 - \|y\|^2 + \left\|\hat{y}\right\|^2\right) \\ -\left(D(\hat{x})u\right)^T\left(c(x) - c(\hat{x}) + \left(D(x) - D(\hat{x})\right)u\right)\end{gathered}$$

$$\leq -\frac{1}{2}\left\|\Theta(x,\hat{x})u - \Theta^{-T}(x,\hat{x})w^T(x,\hat{x})\right\|^2 + \frac{\gamma^2}{2}\|u\|^2 - \frac{1}{2}\|r\|^2$$
$$\leq \frac{\gamma^2}{2}\|u\|^2 - \frac{1}{2}\|r\|^2. \tag{10.34}$$

Thus, by adopting the evaluation window $[0, \tau]$, we finally have

$$\|r_\tau\|_2^2 \leq \gamma^2\|u_\tau\|_2^2 + 2V(x(0), \hat{x}(0)),$$

which completes the proof.

Theorem 10.3 provides us with an algorithm for the design of an $\mathcal{L}_2$-NFDF. It consists of

- solving HJI (10.31) for $V(x, \hat{x})$ and
- solving (10.32) for $L(\hat{x})$.

It is worth noticing that the solvability of (10.31) and (10.32) leads to (10.34), which means that the affine system (10.2) is weakly output re-constructible, as given in Definition 10.4. If they are solvable, the following FD scheme can be applied:

- Run the residual generator (10.23)–(10.24);
- Set the adaptive threshold

$$J_{th} = \gamma^2\|u_\tau\|_2^2 + 2V(x(0), \hat{x}(0));$$

- Define the decision logic (10.27).

10.3.3 An Extension to $\mathcal{L}_2$-RNFDF Design

With a slight modification, the major result in Theorem 10.3 can be applied to solving the following (robust) FD problem.

Consider the nonlinear system of the form

$$\Sigma_d : \begin{cases} \dot{x} = a(x) + E_d(x)d, \\ y = c(x) + F_d(x)d, \end{cases} \tag{10.35}$$

where d is the unknown input vector and $\mathcal{L}_2$-bounded with

$$\|d_\tau\|_2 \leq \delta_d. \tag{10.36}$$

Definition 10.6 *Given the nonlinear system (10.35), the NFDF of the form*

$$\dot{\hat{x}} = a(\hat{x}) + L(\hat{x})\left(y - c(\hat{x})\right), r = y - c(\hat{x}) \tag{10.37}$$

is called $\mathcal{L}_2$ robust NFDF (RNFDF) if for some constant $\gamma > 0$

$$\|r_\tau\|_2^2 \leq \gamma^2 \delta_d^2 + \gamma_o, \tag{10.38}$$

where $\gamma_o \geq 0$ is a (finite) constant for given $x(0), \hat{x}(0)$.

Let

$$\bar{f}(x, \hat{x}) = \begin{bmatrix} a(x) \\ a(\hat{x}) \end{bmatrix}, G(x, \hat{x}) = \begin{bmatrix} E_d(x) \\ 0 \end{bmatrix}.$$

We have the following theorem.

Theorem 10.4 *Consider the system (10.35) and the NFDF (10.37). Assume that*

- $$\gamma^2 I - F_d^T(x) F_d(x) > 0,$$

  and define

  $$\gamma^2 I - F_d^T(x) F_d(x) = \Theta_d^T(x, \hat{x}) \Theta_d(x, \hat{x}),$$

- *there exists $V(x, \hat{x}) \geq 0$ such that the HJI*

  $$\begin{aligned} &V_{x,\hat{x}}(x, \hat{x}) \bar{f}(x, \hat{x}) + \frac{1}{2}\left(c^T(x)c(x) - c^T(\hat{x})c(\hat{x})\right) \\ &+\frac{1}{2} w_d(x, \hat{x}) \left(\Theta_d^T(x, \hat{x}) \Theta_d(x, \hat{x})\right)^{-1} w_d^T(x, \hat{x}) \leq 0, \quad (10.39) \\ &w_d(x, \hat{x}) = V_{x,\hat{x}}(x, \hat{x}) G(x, \hat{x}) + c^T(x) F_d(x), \quad (10.40) \end{aligned}$$

  is solvable for $V_{x,\hat{x}}(x, \hat{x})$, and
- *$L(\hat{x})$ solves*

  $$V_{\hat{x}}(x, \hat{x}) L(\hat{x}) = c^T(\hat{x}). \tag{10.41}$$

Then, it holds

$$\|r_\tau\|_2^2 \leq \gamma^2 \delta_d^2 + 2V(x(0), \hat{x}(0)). \tag{10.42}$$

The proof of this theorem is similar to Theorem 10.3 and is thus omitted.

Based on (10.42), corresponding to the evaluation function

$$J = \|r_\tau\|_2^2,$$

the threshold J_{th} can then be set as

$$J_{th} = \gamma^2 \delta_d^2 + 2V(x(0), \hat{x}(0)). \tag{10.43}$$

10.3.4 On FD Schemes for $\mathcal{L}_2$-stable Affine Systems

We now consider the NFDF design problem for $\mathcal{L}_2$-stable affine systems. Although this is a special case of our previous study, it is helpful to gain a deeper insight into the addressed nonlinear FD problems.

Recall that for the $\mathcal{L}_2$-stable system (10.2) it holds, for some $\bar{\gamma} > 0$, $\bar{\gamma}_o > 0$,

$$\|y_\tau\|_2 \le \bar{\gamma}\, \|u_\tau\|_2 + \bar{\gamma}_o,$$

which can be further written as, for some $\gamma_u > 0$, $\gamma_o > 0$,

$$\|y_\tau\|_2^2 \le \gamma_u^2\, \|u_\tau\|_2^2 + \gamma_o. \tag{10.44}$$

This means that the original system (10.2) itself can also serve as an NFDF. Using the process input and output variables u, y, the threshold setting (10.25) and detection logic (10.27), an FD system is then built. On the other hand, since γ_u^2 can be (very) large, this leads to a high threshold setting. Consequently, the fault detectability may become poor. To illustrate this fact, consider a simple case with a sensor fault modelled by

$$y = c(x) + D(x)u + w,$$

where w denotes the sensor fault vector. According to the detection logic (10.27), w is only detectable if

$$\int_0^\tau \|c(x) + D(x)u + w\|^2\, dt > J_{th},$$
$$J_{th} = \gamma_u^2 \int_0^\tau \|u\|^2\, dt + \sup \gamma_o. \tag{10.45}$$

It is evident that a large γ_u^2 means that only large w can be detected. In other words, the fault detectability is, in this case, poor.

In order to improve the FD performance, we now apply a simple residual generator defined by

$$\dot{\hat{x}} = a(\hat{x}) + B(\hat{x})u, r = y - c(\hat{x}) - D(\hat{x})u. \tag{10.46}$$

Let $\gamma > 0$ ensure

$$\gamma^2 I - \left(D(x) - D(\hat{x})\right)^T \left(D(x) - D(\hat{x})\right) > 0,$$

and set

$$\gamma^2 I - \left(D(x) - D(\hat{x})\right)^T \left(D(x) - D(\hat{x})\right) = \Theta^T(x, \hat{x})\Theta(x, \hat{x}).$$

If there exists $V(x,\hat{x}) \geq 0$ that solves the HJI

$$\begin{gathered} V_x(x,\hat{x})a(x) + V_{\hat{x}}(x,\hat{x})a(\hat{x}) + \frac{1}{2}\left\|c(x)-c(\hat{x})\right\|^2 \\ +\frac{1}{2}w(x,\hat{x})\left(\Theta^T(x,\hat{x})\Theta(x,\hat{x})\right)^{-1}w^T(x,\hat{x}) \leq 0, \qquad (10.47) \\ w(x,\hat{x}) = V_x(x,\hat{x})B(x) + V_{\hat{x}}(x,\hat{x})B(\hat{x}) + \left(c(x)-c(\hat{x})\right)^T\left(D(x)-D(\hat{x})\right), \end{gathered}$$

then we have

$$\|r_\tau\|_2^2 \leq \gamma^2\|u_\tau\|_2^2 + 2V(x(0),\hat{x}(0)). \qquad (10.48)$$

The proof of (10.48) is straightforward and can be done as follows. Consider

$$\begin{gathered} \dot{V}(x,\hat{x}) = V_x(x,\hat{x})a(x) + V_{\hat{x}}(x,\hat{x})a(\hat{x}) + \left(V_x(x,\hat{x})B(x) + V_{\hat{x}}(x,\hat{x})B(\hat{x})\right)u, \\ \frac{\|r\|^2}{2} = \frac{1}{2}\left\|c(x)-c(\hat{x})\right\|^2 + \left(c(x)-c(\hat{x})\right)^T\left(D(x)-D(\hat{x})\right)u \\ +\frac{1}{2}\left\|\left(D(x)-D(\hat{x})\right)u\right\|^2, \\ \frac{1}{2}\left\|\Theta(x,\hat{x})u - \Theta^{-T}(x,\hat{x})w^T(x,\hat{x})\right\|^2 = \frac{\gamma^2}{2}\|u\|^2 - \frac{1}{2}\left\|\left(D(x)-D(\hat{x})\right)u\right\|^2 \\ -w(x,\hat{x})u + \frac{1}{2}w(x,\hat{x})\left(\Theta^T(x,\hat{x})\Theta(x,\hat{x})\right)^{-1}w^T(x,\hat{x}). \end{gathered}$$

It turns out

$$\begin{gathered} \frac{\|r\|^2}{2} = \frac{\gamma^2}{2}\|u\|^2 + \frac{1}{2}\left\|c(x)-c(\hat{x})\right\|^2 - \left(V_x(x,\hat{x})B(x) + V_{\hat{x}}(x,\hat{x})B(\hat{x})\right)u + \\ \frac{1}{2}w(x,\hat{x})\left(\Theta^T(x,\hat{x})\Theta(x,\hat{x})\right)^{-1}w^T(x,\hat{x}) - \frac{1}{2}\left\|\Theta(x,\hat{x})u - \Theta^{-T}(x,\hat{x})w^T(x,\hat{x})\right\|^2 \\ \Longrightarrow \dot{V}(x,\hat{x}) \leq -\frac{\|r\|^2}{2} + \frac{\gamma^2}{2}\|u\|^2. \end{gathered}$$

It yields

$$\frac{\|r_\tau\|_2^2}{2} + V(x,\hat{x}) \leq \frac{\gamma^2\|u_\tau\|_2^2}{2} + V(x(0),\hat{x}(0)),$$

and finally we have (10.48). It follows immediately from (10.48) that the threshold setting in this case is

$$J_{th} = \gamma^2\|u_\tau\|_2^2 + \gamma_o, \gamma_o = \max_{x(0),\hat{x}(0)} 2V(x(0),\hat{x}(0)). \qquad (10.49)$$

We now compare the FD scheme based on (10.48) and the one given in (10.44). Recall that for the system (10.2), (10.44) holds only if $\gamma_u^2 I - D^T(x)D(x) > 0$, and, moreover, the following HJI is solvable for given γ_u

$$\begin{gathered}
V_x(x,\hat{x})a(x) + V_{\hat{x}}(x,\hat{x})a(\hat{x}) + \frac{1}{2}\left\| c(x) \right\|^2 \\
+\frac{1}{2} w(x,\hat{x}) \left(\Theta^T(x,\hat{x})\Theta(x,\hat{x}) \right)^{-1} w^T(x,\hat{x}) \leq 0, \\
w(x,\hat{x}) = V_x(x,\hat{x})B(x) + V_{\hat{x}}(x,\hat{x})B(\hat{x}) + c^T(x)D(x), \\
\gamma_u^2 I - D^T(x)D(x) = \Theta^T(x,\hat{x})\Theta(x,\hat{x}).
\end{gathered} \tag{10.50}$$

Assume that $\hat{x}$ is a good estimate of x in the sense of

$$x - \hat{x} \in \mathcal{B}_\delta$$

for some δ and ensures that $\forall x - \hat{x} \in \mathcal{B}_\delta$,

$$\begin{aligned}
\left(D(x) - D(\hat{x}) \right)^T \left(D(x) - D(\hat{x}) \right) &< D^T(x)D(x), \\
\left\| c(x) - c(\hat{x}) \right\|^2 &< \left\| c(x) \right\|^2,
\end{aligned}$$

and, moreover, for a γ^2 smaller than γ_u^2, the HJI (10.47) is solvable. Then, comparing the thresholds (10.45) and (10.49) makes it clear that

$$\gamma^2 \int_0^\tau \left\| u \right\|^2 dt < \gamma_u^2 \int_0^\tau \left\| u \right\|^2 dt.$$

That means for a large $\int_0^\tau \left\| u \right\|^2 dt$, threshold setting (10.49) is lower than the one given by (10.45). On the other hand, since

$$\hat{y} = c(\hat{x}) + D(\hat{x})u$$

is independent of any fault, the influence of a fault vector, for instance the sensor fault, on the residual vector is identical with the one on y. As a result, the FD performance of the residual generator (10.46) is improved in comparison with an FD system based on a direct use of the process input and output variables.

The above discussion is of practical interest, since in many automatic control systems, the plant is stable and a parallel running model is embedded in the system for monitoring or control purpose. The so-called internal model control (IMC) system is a typical example. By means of the above detection scheme, an $\mathcal{L}_2$-NFDF with satisfactory FD performance can be realised without additional online computation and engineering costs.

10.4 Examples

We now illustrate the main results presented in the previous sections by some examples.

Example 10.1 *The first example is adopted from the literature, which is given at the end of this chapter, and used to illustrate the result given in Theorem 10.1. Consider the nonlinear system (10.1) with*

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, f(x,u) = \begin{bmatrix} -x_1^3 + x_2 \\ -x_2^3 + u \end{bmatrix}, h(x,u) = x_1. \tag{10.51}$$

Suppose that $u \in \mathcal{U} = [-1, 1]$. *It is given in the literature that for every initial condition* $x(0) \in \mathcal{R}^2$ *and* $u \in \mathcal{U} = [-1, 1]$, *the solution* $x(t)$ *of (10.51) enters the compact set*

$$S = \{x \in \mathcal{R}^2 : W(x) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 \leq \sqrt{10}/2\}. \tag{10.52}$$

For our purpose, we now construct the observer-based residual generator as follows

$$\begin{aligned} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} &= \phi(\hat{x}, u, y) = \begin{bmatrix} -\hat{x}_1^3 + \hat{x}_2 \\ -\hat{x}_2^3 + u \end{bmatrix} + \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} (y - \hat{x}_1), \\ r &= y - \hat{x}_1. \end{aligned}$$

Let

$$V(x, \hat{x}) = \frac{1}{2}(x - \hat{x})^T P (x - \hat{x}), P = \begin{bmatrix} 1 & -a \\ -a & b \end{bmatrix}$$

for some a, b *ensuring* $P > 0$, *and denote*

$$e_1 = x_1 - \hat{x}_1, e_2 = x_2 - \hat{x}_2.$$

Next, assume that

$$x_1^2 + x_2^2 \leq c, \hat{x}_1^2 + \hat{x}_2^2 \leq d$$

for some c, d. *It is straightforward to verify*

$$\begin{aligned} &V_x(x, \hat{x}) f(x,u) + V_{\hat{x}}(x, \hat{x}) \phi(\hat{x}, u, y) \\ &\leq \left(\frac{9a}{8}(c+d)^2 - (l_1 - al_2) \right) e_1^2 - \frac{a}{2} e_2^2 + (1 + al_1 - bl_2) e_1 e_2. \end{aligned}$$

Now, select l_1 *and* l_2 *such that*

$$\frac{9a}{8}(c+d)^2 - l_1 + al_2 + \frac{a}{2} = 0, 1 + al_1 - bl_2 = 0,$$

for a, b, d satisfying $b > a^2$, $a > 0$ and $d > c$. As a result, it holds

$$\frac{\lambda_1}{2} r^T r \leq V(x, \hat{x}) \leq \frac{\lambda_2}{2}(x - \hat{x})^T (x - \hat{x}),$$
$$V_x(x, \hat{x}) f(x, u) + V_{\hat{x}}(x, \hat{x}) \phi(\hat{x}, u, y) \leq -\frac{a}{2}(x - \hat{x})^T (x - \hat{x}),$$

where λ_1, λ_2 ($\lambda_2 \geq \lambda_1$) are eigenvalues of P, which means, according to Definition 10.3, system (10.51) is output re-constructible. Moreover, since

$$\dot{V}(x, \hat{x}) \leq -\frac{a}{\lambda_2} V(x, \hat{x}),$$

it is known that

$$V(x, \hat{x}) \leq e^{-\frac{a}{\lambda_2} t} V(x_0, \hat{x}_0),$$

which leads to

$$\|r(t)\|^2 \leq \frac{2}{\lambda_1} e^{-\frac{a}{\lambda_2} t} V(x_0, \hat{x}_0) =: \beta \left(\left\| x(0) - \hat{x}(0) \right\|, t \right)$$

This illustrates the result given in Theorem 10.1.

Example 10.2 *This example demonstrates the application of Theorem 10.3 for the design of an $\mathcal{L}_2$-NFDF. Consider the following affine system*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 - \sin^3 x_1 \\ -a \sin x_1 \end{bmatrix} + \begin{bmatrix} \sin x_1 \\ 0 \end{bmatrix} u, \; y = \sin^2 x_1$$

with $a > \frac{1}{2}$. Construct the observer-based residual generator

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} \hat{x}_2 - \sin^3 \hat{x}_1 \\ -a \sin \hat{x}_1 \end{bmatrix} + \begin{bmatrix} \sin \hat{x}_1 \\ 0 \end{bmatrix} u + L(\hat{x}) r,$$
$$r = y - \hat{y} = \sin^2 x_1 - \sin^2 \hat{x}_1.$$

Thus, the functions in the system (10.28) and Theorem 10.3 are given by

$$\bar{f}(x, \hat{x}) = \begin{bmatrix} x_2 - \sin^3 x_1 \\ -a \sin x_1 \\ \hat{x}_2 - \sin^3 \hat{x}_1 \\ -a \sin \hat{x}_1 \end{bmatrix}, G(x, \hat{x}) = \begin{bmatrix} \sin x_1 \\ 0 \\ \sin \hat{x}_1 \\ 0 \end{bmatrix},$$
$$c(x) = \sin^2 x_1, c(\hat{x}) = \sin^2 \hat{x}_1, D(x) = D(\hat{x}) = 0,$$
$$\Theta^T(x, \hat{x}) \Theta(x, \hat{x}) = \gamma^2,$$

for some $\gamma > 0$. Let

$$V(x,\hat{x}) = a(1-\cos x_1) + \frac{1}{2}x_2^2 + a(1-\cos\hat{x}_1) + \frac{1}{2}\hat{x}_2^2 \Longrightarrow$$
$$V_{x,\hat{x}}(x,\hat{x}) = \begin{bmatrix} a\sin x_1 & x_2 & a\sin\hat{x}_1 & \hat{x}_2 \end{bmatrix},$$
$$w(x,\hat{x}) = V_{x,\hat{x}}(x,\hat{x})G(x,\hat{x}) = a\sin^2 x_1 + a\sin^2\hat{x}_1.$$

It turns out

$$\begin{aligned}
&V_{x,\hat{x}}(x,\hat{x})\bar{f}(x,\hat{x}) + \frac{1}{2}c^T(x)c(x) - \frac{1}{2}c^T(\hat{x})c(\hat{x}) + \frac{1}{2\gamma^2}w^2(x,\hat{x}) \\
&= \left(-a + \frac{1}{2} + \frac{a^2}{2\gamma^2}\right)\sin^4 x_1 + \frac{a^2}{\gamma^2}\sin^2 x_1 \sin^2\hat{x}_1 + \left(-a - \frac{1}{2} + \frac{a^2}{2\gamma^2}\right)\sin^4\hat{x}_1 \\
&= -\frac{a^2}{2\gamma^2}(\sin^2 x_1 - \sin^2\hat{x}_1)^2 + \left(-a + \frac{1}{2} + \frac{a^2}{\gamma^2}\right)\sin^4 x_1 + \left(-a - \frac{1}{2} + \frac{a^2}{\gamma^2}\right)\sin^4\hat{x}_1 \\
&\le \left(-a + \frac{1}{2} + \frac{a^2}{\gamma^2}\right)\sin^4 x_1 + \left(-a - \frac{1}{2} + \frac{a^2}{\gamma^2}\right)\sin^4\hat{x}_1.
\end{aligned}$$

It is evident that for

$$\gamma \ge \frac{a}{\sqrt{a - \frac{1}{2}}},$$

it holds

$$-a + \frac{1}{2} + \frac{a^2}{\gamma^2} \le 0, -a - \frac{1}{2} + \frac{a^2}{\gamma^2} \le 0.$$

As a result, the HJI (10.31) is satisfied. Next, notice that

$$L(\hat{x}) = \begin{bmatrix} \frac{1}{a}\sin\hat{x}_1 \\ 0 \end{bmatrix}$$

solves

$$V_{\hat{x}}(x,\hat{x})L(\hat{x}) = \sin^2\hat{x}_1.$$

Hence, the NFDF can be constructed as

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} \hat{x}_2 - \sin^3\hat{x}_1 + u\sin\hat{x}_1 + \frac{\sin\hat{x}_1}{a}(\sin^2 x_1 - \sin^2\hat{x}_1) \\ -a\sin\hat{x}_1 \end{bmatrix},$$
$$r = \sin^2 x_1 - \sin^2\hat{x}_1.$$

In order to verify the $\mathcal{L}_2$-stability of the NFDF, we choose the input function shown in Fig. 10.1. The simulation results in Fig. 10.2 show that the output signal $r(t)$ of the NFDF is $\mathcal{L}_2$-bounded in the fault-free case with $a = 0.6$ and $x(0) = (1, -0.3), \hat{x}(0) = (-1, -0.2)$. For the FD purpose, we choose a constant sensor fault 0.7 *occurred at* 60 sec. *With residual evaluation and threshold compu-*

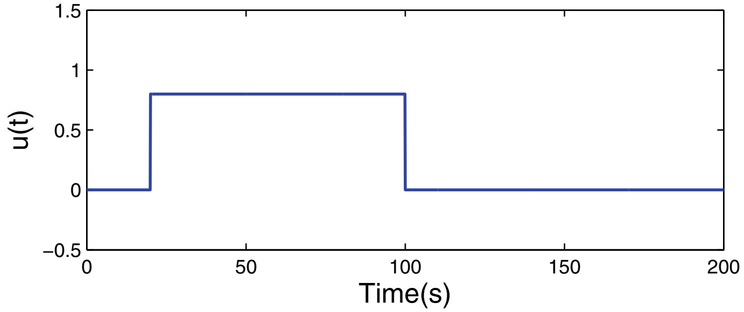


Fig. 10.1 Input signal $u(t)$

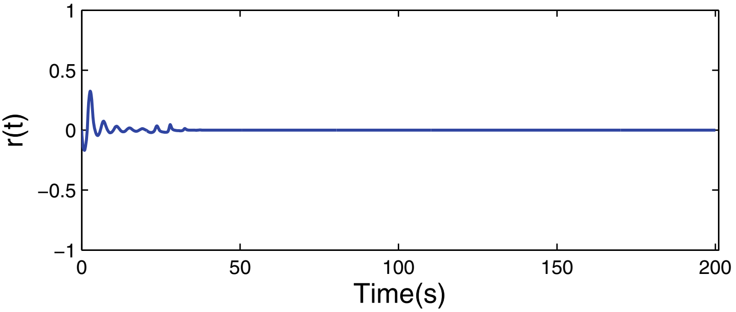


Fig. 10.2 Residual signal $r(t)$

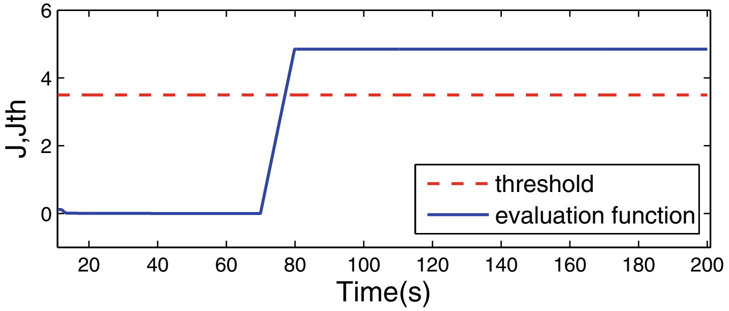


Fig. 10.3 Detection of a sensor fault

tation method provided in Theorem 10.3, it is evident that the fault can be detected as shown in Fig. 10.3.

Example 10.3 *In this example, we demonstrate the results of Theorem 10.4 for the design of an $\mathcal{L}_2$-RNFDF and its application in fault detection. Consider the system described by*

$$\begin{aligned}\dot{x}_1 &= -x_2 - 2x_1^3 + \frac{1}{2}x_1 d,\\ \dot{x}_2 &= x_1 - x_2^3 - 2x_1^2 x_2,\\ y &= x_1^2 + x_2^2.\end{aligned}$$

To design an $\mathcal{L}_2$-RNFDF for the above system, we propose

$$V\left(x, \hat{x}\right) = x_1^2 + x_2^2 + \hat{x}_1^2 + \hat{x}_2^2.$$

It can be verified that the HJI (10.39) is satisfied for $\gamma \geq \frac{1}{2}$. According to (10.41), the observer gain matrix is given by

$$L(\hat{x}) = \frac{1}{2}\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix},$$

which leads to the $\mathcal{L}_2$-RNFDF of the form

$$\begin{aligned}\dot{\hat{x}}_1 &= -\hat{x}_2 - 2\hat{x}_1^3 + \frac{1}{2}\hat{x}_1 r,\\ \dot{\hat{x}}_2 &= \hat{x}_1 - \hat{x}_2^3 - 2\hat{x}_1^2\hat{x}_2 + \frac{1}{2}\hat{x}_2 r,\\ r &= x_1^2 + x_2^2 - \hat{x}_1^2 - \hat{x}_2^2.\end{aligned}$$

Example 10.4 *In this example, we compare two different fault detection schemes as discussed in Sub-section10.3.4 and demonstrate that a residual generator delivers better FD performance in comparison with an FD system based on a direct use of the process input and output variables.*

Consider the system described by

$$\dot{x}_1 = -x_1^3 + \frac{1}{2}x_1 u, \dot{x}_2 = -x_2 - x_2^3, y = x_2 + 2u + w, \tag{10.53}$$

where w represents a sensor fault.

First, it is demonstrated that the above system is $\mathcal{L}_2$-stable, and then (10.44) is applied for the FD performance. To this end, let

$$V(x) = x_1^2 + x_2^2.$$

It can be proved that the HJI (10.50) is satisfied for $\gamma_u > \sqrt{6}$, which leads to

$$\|y_\tau\|_2^2 \leq 6\|u_\tau\|_2^2 + \gamma_o.$$

Assume that $\|x(0)\| \leq 0.5$. It is then reasonable to set

$$J_{th} = 6 \int_{t_0}^{t_0+\tau} \|u\|^2 \, dt + 1. \tag{10.54}$$

Next, we consider residual generator

$$\dot{\hat{x}}_1 = -\hat{x}_1^3 + \frac{1}{2}\hat{x}_1 u, \dot{\hat{x}}_2 = -\hat{x}_2 - \hat{x}_2^3, r = y - \hat{x}_2 - 2u, \tag{10.55}$$

and determine γ for the threshold setting given in (10.48) **.** *Let*

$$V(x, \hat{x}) = x_1^2 + x_2^2 + \hat{x}_1^2 + \hat{x}_2^2.$$

The HJI (10.47) holds for $\gamma > \frac{\sqrt{2}}{2}$. Assume that

$$\|x(0)\| \leq 1, \left\|\hat{x}(0)\right\| = 0.$$

The corresponding threshold is set to be

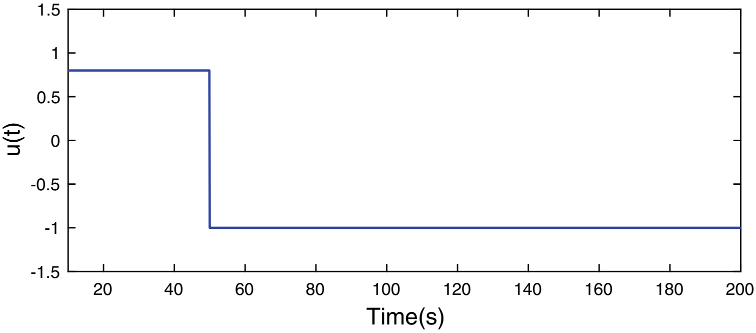


Fig. 10.4 Input signal $u(t)$

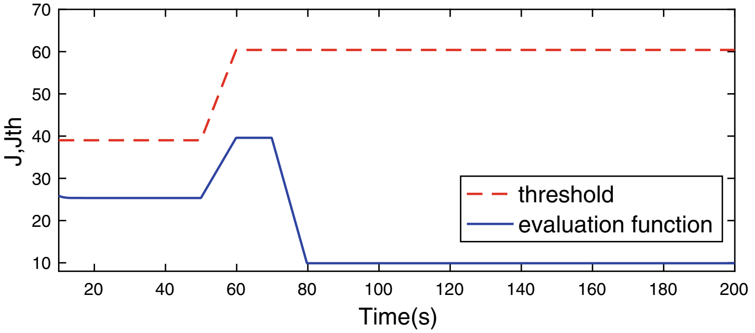


Fig. 10.5 Fault detection based on y and u

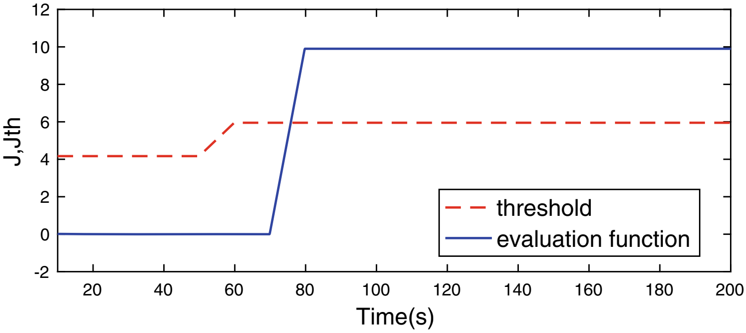


Fig. 10.6 Fault detection based on residual r

$$J_{th} = \frac{1}{2} \int_{t_0}^{t_0+\tau} \|u\|^2 \, dt + 2. \tag{10.56}$$

Comparing the two threshold setting laws, (10.54) and (10.56), makes it clear that the FD performance delivered by the FD system (10.55) with threshold (10.56) is (much) better than the one achieved by the FD system (10.53) and (10.54). In order to demonstrate this, a constant sensor fault $w = 1$ is considered and added in the simulation at $t = 70\,\text{sec}$. We choose $\tau = 10\,\text{sec}$ and the input function $u(t)$ shown in Fig. 10.4. It can be seen from Fig. 10.5 that, with the evaluation based on y and threshold computation by means of (10.54), the fault cannot be detected, while a fault detection based on residual evaluation of FD system (10.55) can be realised as shown in Fig. 10.6. The initial conditions are

$$x(0) = \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}, \hat{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

10.5 Notes and References

One of the most challenging topics in the FD research and application areas is nonlinear observer-based FDI. The review in [1] shows that the application of nonlinear observer theory built the main stream in the nonlinear observer-based FD study since 1990s. While the first studies have been mainly devoted to the application of feedback-based linearisation and differential algebra techniques to observer-based residual generator design [2–4], and the geometric approach to nonlinear FD [5], the recent research efforts address systems with a special class of nonlinearities, typically Lipschitz nonlinearity [6, 7], sector bounded nonlinearity [8] or special types of control systems like nonlinear NCSs (networked control systems) [9]. Differently, [10–12] have investigated residual evaluation, threshold setting and residual generator optimisation issues for nonlinear observer-based FD systems. It can be observed that

- only few of these studies have dealt with residual generator and evaluation as well as decision making in an integrated way, and
- most of efforts have been made on the FD system design but only few on analysis issues.

In recent years, much attention has been paid to the application of those techniques to addressing nonlinear FD issues, which are newly established for dealing with analysis and synthesis of nonlinear dynamic systems more efficiently. For instance, fuzzy technique based FD [13–16], adaptive fault diagnosis for nonlinear systems [17, 18], LPV-based FD [19, 20] or sliding mode observer-based fault detection [21, 22] have been reported.

Having noticed that little attention has been paid to the existence conditions of nonlinear observer-based FD systems and there is no commonly used conditions for checking the existence of an observer-based FD system for nonlinear systems, we have, in collaboration with Prof. Yang and Dr. Li, made considerable initial efforts of approaching these issues. The first results of this work have been summarised in [23, 24], which build the core of this and the next chapters.

Our collaborative work has been considerably inspired by the study on the input-output stability and stabilisation in the past decades [25–28]. We notice that rich results have been published at that time, and some of them are very helpful for our tasks described above. For instance, the concept of weak detectability proposed in [25] has been applied for our study on the existence conditions of FD systems, while $\mathcal{L}_2$ -stability theory [28] is the major tool for our study on the integrated design of FD systems.

In this chapter, observer-based FD issues for nonlinear systems are addressed. For the purpose of determining the existence of an observer-based FD system, consisting of an observer-based residual generator, residual evaluation and decision making, we have first introduced

- two types of observer-based nonlinear FD systems, the $\mathcal{L}_\infty$ and $\mathcal{L}_2$ observer-based FD systems, and
- the concepts of output re-constructability as well as weak output re-constructability.

The concept weak detectability was proposed in [25] and is widely used in the study on the stabilisation of nonlinear systems by output feedback [27, 29].

It is proved that if a nonlinear system is output re-constructible, then an $\mathcal{L}_\infty$ observer-based FD system exists. For constructing an $\mathcal{L}_2$ observer-based FD system, the weakly output re-constructability is sufficient. As remarked, a similar proof of Theorem 10.1 can be found in [30, 31] using IOS-Lyapunov function.

In the second part of our work, an integrated design scheme for affine nonlinear systems with the aid of $\mathcal{L}_2$-stability theory is proposed and applied for investigating FD issues for systems with unknown inputs and $\mathcal{L}_2$-stable systems.

We have included four academic examples to illustrate the major theoretical results. The first example, Example 10.1, is adopted from [32].

References

1. E. Alcorta-Garcia and P. Frank, “Deterministic nonlinear observer based approaches to fault diagnosis: A survey,” *Control Engineering practice*, vol. 5(5), pp. 663–670, 1997.
2. R. Seliger and P. Frank, “Fault diagnosis by disturbance decoupled nonlinear observers,” in *Proceedings of the CDC91*, Brighton, England, 1991, pp. 2248–2253.
3. H. Hammouri, M. Kinnaert, and E. E. Yaagoubi, “Observer-based approach to fault detection and isolation for nonlinear systems,” *IEEE Trans. on Automatic Control*, vol. 44(10), pp. 1879–1884, 1999.
4. P. Kabore and H. Wang, “Design of fault diagnosis filters and fault-tolerant control for a class of nonlinear systems,” *IEEE Trans. on Autom. Control*, vol. 46, pp. 1805–1810, 2001.
5. C. D. Persis and A. Isidori, “A geometric approach to nonlinear fault detection and isolation,” *IEEE Trans. on Autom. Control*, vol. 46, pp. 853–865, 2001.
6. A. M. Pertew, H. J. Marquez, and Q. Zhao, “LMI-based sensor fault diagnosis for nonlinear Lipschitz systems,” *Automatica*, vol. 43, pp. 1464–1469, 2007.
7. X. Zhang, M. M. Polycarpou, and T. Parisini, “Fault diagnosis of a class of nonlinear uncertain systems with Lipschitz nonlinearities using adaptive estimation,” *Automatica*, vol. 46, pp. 290–299, 2010.
8. X. He, Z. Wang, and D. Zhou, “Robust hinf filtering for time-delay systems with probabilistic sensor faults,” *IEEE Signal Process. Lett.*, vol. 16, pp. 442–445, 2009.
9. Z. Mao, B. Jiang, and P. Shi, “Protocol and fault detection design for nonlinear networked control systems,” *IEEE Trans. on Circuits and Syst. II: express briefs*, vol. 56, pp. 255–259, 2009.
10. M. Abid, W. Chen, S. X. Ding, and A. Q. Khan, “Optimal residual evaluation for nonlinear systems using post-filter and threshold,” *Int. J. Contr.*, vol. 84, pp. 526–539, 2011.
11. A. Q. Khan and S. X. Ding, “Threshold computation for fault detection in a class of discrete-time nonlinear systems,” *Int. J. Adapt. Control Signal Process.*, vol. 25, pp. 407–429, 2011.
12. A. Q. Khan, M. Abid, and S. X. Ding, “Fault detection filter design for discrete-time nonlinear systems—a mixed H_- and H_{inf} optimization,” *Syst. Contr. Lett.*, vol. 67, pp. 46–54, 2014.
13. S. K. Nguang, P. Shi, and S. Ding, “Fault detection for uncertain fuzzy systems: An LMI approach,” *IEEE Trans. on Fuzzy Systems*, vol. 15, pp. 1251–1262, 2007.
14. M. Chadli, A. Abdo, and S. X. Ding, “H_-/H_{inf} fault detection filter design for discrete-time takagi-sugeno fuzzy system,” *Automatica*, vol. 49, pp. 1996–2005, 2013.
15. L. Li, S. X. Ding, J. Qui, Y. Yang, and Y. Zhang, “Weighted fuzzy observer-based fault detection approach for discrete-time nonlinear systems via piecewise-fuzzy lyapunov functions,” *IEEE Trans. on Fuzzy Systems*, vol. 24, pp. 1320–1333, 2016.
16. L. Li, S. X. Ding, J. Qui, Y. Yang, and D. Xu, “Fuzzy observer-based fault detection design approach for nonlinear processes,” *IEEE Trans. on Syst., Man, and Cybernetics: Systems*, vol. 47, pp. 1941–1952, 2017.
17. A. Xu and Q. Zhang, “Nonlinear system fault diagnosis based on adaptive estimation,” *Automatica*, vol. 40, pp. 1181–1193, 2004.
18. X. Zhang, M. M. Polycarpou, and T. Parisini, “Adaptive fault diagnosis and fault-tolerant control of MIMO nonlinear uncertain systems,” *Int. J. of Contr.*, vol. 83, pp. 1054–1080, 2010.
19. J. Bokor and G. Balas, “Detection filter design for LPV systems—a geometric approach,” *Automatica*, vol. 40, pp. 511–518, 2004.
20. A. Armeni, A. Casavola, and E. Mosca, “Robust fault detection and isolation for LPV systems under a sensitivity constraint,” *Int. J. Adapt. Control and Signal process.*, vol. 23, pp. 55–72, 2009.
21. T. Floquet, J. Barbot, W. Perruquetti, and M. Djemai, “On the robust fault detection via sliding mode disturbance observer,” *Int. J. Control*, vol. 77, pp. 622–629, 2004.
22. X.-G. Yan and C. Edwards, “Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems,” *Int. J. Syst. Sci.*, vol. 39, pp. 349–359, 2008.
23. Y. Yang, S. X. Ding, and L. Li, “On observer-based fault detection for nonlinear systems,” *Syst. Contr. Lett.*, vol. 82, pp. 1399–1410, 2015.

24. Y. Yang, S. X. Ding, and L. Li, “Parametrization of nonlinear observer-based fault detection systems,” *IEEE Trans. on Automatic Control*, vol. 61, pp. 3687–3692, 2016.
25. M. Vidyasagar, “On the stabilization of nonlinear systems using state detection,” *IEEE Trans. on Automat. Control*, vol. 25, pp. 504–509, 1980.
26. E. D. Sontag and Y. Wang, “On characterizations of the input-to-state stability property,” *Syst. Contr. Lett.*, vol. 24, pp. 351–359, 1995.
27. W.-M. Lu, “A state-space approach to parameterization of stabilizing controllers for nonlinear systems,” *IEEE Trans. on Automatic Control*, vol. 40, pp. 1576–1588, 1995.
28. A. Van der Schaft, *L2—Gain and Passivity Techniques in Nonlinear Control*. London: Springer, 2000.
29. D.-J. Pan, Z.-Z. Han, and Z.-J. Zhang, “Bounded-input-bounded-output stabilization of nonlinear systems using state detectors,” *Systems and Control Letters*, vol. 21, pp. 189–198, 1993.
30. A.-R. Teel and L. Praly, “A smooth Lyapunov function from a class-KL estimate involving two positive semidefinite functions,” *Control, Optimisation and Calculus of Variations*, vol. 29, pp. 313–367, 2000.
31. E. Sontag and Y. Wang, “Characterization of input to output stability,” *SIAM. J. Control Optim.*, vol. 39, pp. 226–249, 2001.
32. I. Karafyllis and C. Kravaris, “Global exponential observers for two classes of nonlinear systems,” *Syst. Contr. Lett.*, vol. 61, pp. 797–806, 2012.

Chapter 11
Parameterisation of Observer-Based Fault Detection Systems

Recall that for a given LTI system modelled by

$$\dot{x} = Ax + Bu, y = Cx + Du,$$

all corresponding LTI residual generators can be parameterised by

$$r(s) = R(s)\left(\hat{M}(s)y(s) - \hat{N}(s)u(s)\right),$$

where $R(s)$ is a stable post-filter and $\hat{M}(s)$, $\hat{N}(s)$ are stable, left coprime and build the LCF of $G(s)$,

$$\hat{M}^{-1}(s)\hat{N}(s) = G(s) = D + C\left(sI - A\right)^{-1}B.$$

Moreover, it holds

$$\hat{M}(s)y(s) - \hat{N}(s)u(s) = y(s) - \hat{y}(s),$$

where $\hat{y}$ is the output estimate delivered by a full-order observer

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}), \hat{y} = C\hat{x} + Du.$$

System parameterisation is essential for system analysis and optimisation. This motivates us to investigate, in this chapter, the parameterisation issues of nonlinear observer-based residual generators and fault detection systems. We consider nonlinear systems described by

$$\Sigma : \dot{x} = f(x, u), y = h(x, u), \tag{11.1}$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}^p$, $y \in \mathcal{R}^m$ denote the state, input and output vectors, respectively. $f(x, u)$ and $h(x, u)$ are continuously differentiable nonlinear functions with appropriate dimensions. System (11.1) is called faulty if undesirable changes in the

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_11

system dynamics are caused by some faults. It is supposed that the faulty system dynamics is modelled by

$$\Sigma_f : \dot{x} = \bar{f}(x, u, w), y = \bar{h}(x, u, w) \tag{11.2}$$

with $w \in \mathcal{R}^q$ denoting the fault vector. We call system (11.1) fault-free if

$$w = 0 \text{ and } \bar{f}(x, u, 0) = f(x, u), \bar{h}(x, u, 0) = h(x, u). \tag{11.3}$$

Remember that the parameterisation of LTI residual generators has been inspired by the well-known Youla parameterisation of all stabilising controllers and achieved by means of the coprime factorisation technique. The parameterisation of stabilizing controllers for nonlinear systems has been extensively investigated in the 90s. As a powerful tool for this work, the factorisation technique and the nonlinear kernel and image representations have been applied. Note that, also in this time period, characterisations of the so-called input-to-output stability (IOS) of nonlinear systems have been intensively studied. Analogue to these works, in this chapter we will first apply the nonlinear factorisation and input-output operator techniques to the configuration study on observer-based residual generators, which leads to a parameterisation of observer-based residual generators in form of a cascade connection of a system kernel representation and a post-filter. Based on a state space realisation of the proposed parameterisation, a characterisation of the overall observer-based FD systems including the residual evaluator and the threshold will then be studied. We will focus on the existence conditions of the FD system parameterisation. To this end, the concept of the IOS and some methods for system input/output stabilisation will be applied.

11.1 Problem Formulation

Inspired by the LTI parameterisation form

$$r(s) = R(s)\left(\hat{M}(s)y(s) - \hat{N}(s)u(s)\right) = R(s)\left(y(s) - \hat{y}(s)\right),$$

our first task is to study if a nonlinear observer-based residual generator can be parameterised by

$$r = \Sigma_Q(y - \hat{y}), \tag{11.4}$$

where $\hat{y}$ will be delivered by a nonlinear FDF and Σ_Q represents a (nonlinear) dynamic system with $y - \hat{y}$ as its input vector. To this end, we will apply the kernel and image representations of nonlinear systems known in the literature. Based on the parameterisation configuration (11.4), our second task is to find the existence conditions for the parameterised residual generator and the associated thresholds with respect to the evaluation functions $J_2(r)$ and $J_E(r)$,

$$J_2(r) = \|r(t)\|_{2,\tau}^2 \text{ or } J_E(r) = \|r(t)\|_\infty^2, \tag{11.5}$$

where the threshold determination is based on

$$J_{th,2} = \sup_{w=0} J_2(r),\ J_{th,E} = \sup_{w=0} J_E(r). \tag{11.6}$$

11.2 Parameterisation of Nonlinear Residual Generators

In this section, we address the parameterisation of residual generators for systems described by (11.1). This work is mainly based on the well-established input-output operator approach. To this end, we will adopt, beside the standard notation and the notations defined in Sub-section 10.1.4, the following notations.

A signal space $\mathcal{U}$ denotes a vector space of functions from a time domain to an Euclidean vector space. $\mathcal{U}^s$ represents the subset of all the bounded signals in $\mathcal{U}$. An operator Σ with an input signal space $\mathcal{U}$, an output signal space $\mathcal{Y}$ and an initial condition $x_0 \in \mathcal{X}_0$ is denoted by $\Sigma^{x_0} : \mathcal{U} \to \mathcal{Y}$. It is said to be stable if

$$\forall x_0, u \in \mathcal{U}^s \Rightarrow \Sigma^{x_0}(u) \in \mathcal{Y}^s.$$

The cascade connection of two systems $\Sigma_1^{\xi_0} : \mathcal{U} \times \mathcal{Y} \to \mathcal{Z}$ and $\Sigma_2^{\varsigma_0} : \mathcal{L} \to \mathcal{U} \times \mathcal{Y}$ is denoted by $\Sigma_1^{\xi_0} \circ \Sigma_2^{\varsigma_0} : \mathcal{L} \to \mathcal{Z}$.

For our purpose, we now introduce some definitions. Let

$$\Sigma^{x_0} : \mathcal{U} \to \mathcal{Y},\ \Sigma_f^{x_0} : \mathcal{U} \times \mathcal{W} \to \mathcal{Y}$$

be the operator of (11.1) and (11.2), respectively, and assume that

$$\Sigma^{x_0}(u) = \Sigma_f^{x_0}\begin{pmatrix} u \\ 0 \end{pmatrix}.$$

Definition 11.1 *Given Σ^{x_0}, $\Sigma_f^{x_0}$, the fault vector $w (\neq 0)$ is said to be detectable if for some u, x_0,*

$$\Sigma^{x_0}(u) \neq \Sigma_f^{x_0}\begin{pmatrix} u \\ w \end{pmatrix}.$$

It is reasonable that in our study only detectable faults are considered.

Definition 11.2 *An operator $R_\Sigma^{\xi_0} : \mathcal{U}^s \times \mathcal{Y}^s \to \mathcal{R}^s$ is called (stable) residual generator if*

$$\begin{cases} \forall u, x_0, \exists \xi_0 \text{ so that } R_{\Sigma}^{\xi_0}\begin{pmatrix} u \\ y \end{pmatrix} = 0 \text{ for } w = 0, \\ R_{\Sigma}^{\xi_0}\begin{pmatrix} u \\ y \end{pmatrix} \neq 0 \text{ for detectable } w \neq 0. \end{cases} \tag{11.7}$$

The output of $R_{\Sigma}^{\xi_0}$,

$$r = R_{\Sigma}^{\xi_0}\begin{pmatrix} u \\ y \end{pmatrix},$$

is called residual vector.

Condition (11.7) means that the residual generator is driven by the process input and output vectors, u, y, and the residual vector r should be zero in the fault-free case. Note that x, ξ may have different dimensions.

Definition 11.3 *A stable kernel representation (SKR) of the operator* $\Sigma^{x_0} : \mathcal{U} \to \mathcal{Y}$ *is an operator* $K_{\Sigma}^{\hat{x}_0} : \mathcal{U}^s \times \mathcal{Y}^s \to \mathcal{Z}^s$ *such that, for any* $\hat{x}_0 = x_0$,

$$K_{\Sigma}^{\hat{x}_0}\begin{pmatrix} u \\ y \end{pmatrix} = z = 0, z \in \mathcal{R}^m. \tag{11.8}$$

A stable image representation (SIR) of the operator $\Sigma^{x_0} : \mathcal{U} \to \mathcal{Y}$ *is an operator* $I_{\Sigma^{x_0}} : \mathcal{L}^s \to \mathcal{U}^s \times \mathcal{Y}^s$ *such that* $\forall x_0, u$ *and the resulting* $y = \Sigma^{x_0}(u)$ *there exists* l *so that*

$$\begin{pmatrix} u \\ y \end{pmatrix} = I_{\Sigma^{x_0}}(l). \tag{11.9}$$

The SKR and SIR defined above can be interpreted as an extension of the LTI SKR and SIR introduced and addressed in the previous chapters. Note that z would be different from zero when $\hat{x}_0 \neq x_0$. For an SKR, z is bounded, that is $z \in \mathcal{Z}^s$. In the subsequent section, we shall present the existence conditions for this case. As a dual form, the SIR means, for any $l \in \mathcal{L}^s$ and initial condition, $I_{\Sigma^{x_0}}(l)$ delivers

$$u \in \mathcal{U}^s, y \in \mathcal{Y}^s, y = \Sigma^{x_0}(u).$$

Notice that by means of an SKR we are able to define an operator for an output observer as follows

$$\hat{Y}_{\Sigma}^{\hat{x}_0} : \hat{y} = y - K_{\Sigma}^{\hat{x}_0}\begin{pmatrix} u \\ y \end{pmatrix}. \tag{11.10}$$

In other words, designing an output observer can be viewed equivalently as a problem of finding an SKR, as we know in the LTI case.

The SKR and SIR of Σ^{x_0} are two alternative description forms of Σ^{x_0}, and both of them are stable operators. It follows directly from SKR and SIR definitions that

$$K_{\Sigma}^{x_0} \circ I_{\Sigma^{x_0}} = 0. \tag{11.11}$$

The following definition is needed for introducing the inverses of $K_{\Sigma}^{\hat{x}_0}$, $I_{\Sigma^{x_0}}$.

Definition 11.4 *The SKR $K_{\Sigma}^{\hat{x}_0}$ is said to be coprime if it has a stable right inverse $K_{\Sigma}^{-} : \mathcal{Z}^s \rightarrow \mathcal{U}^s \times \mathcal{Y}^s$ satisfying*

$$K_{\Sigma}^{\hat{x}_0} \circ K_{\Sigma}^{-} = I. \tag{11.12}$$

Analogue to it, the SIR $I_{\Sigma^{x_0}}$ is said to be coprime if it has a stable left inverse $I_{\Sigma}^{-} : \mathcal{U}^s \times \mathcal{Y}^s \rightarrow \mathcal{L}^s$ satisfying

$$I_{\Sigma}^{-} \circ I_{\Sigma^{x_0}} = I. \tag{11.13}$$

We would like to remark that the definition of the SKR is known in the literature. The definition of the SIR is a dual form, which is closely related to the definition of right coprime factorisation of a nonlinear operator (system), also known in the literature. In the sequel, SKR will be applied for our study on the configuration of observer-based residual generators. In the next section, the existence condition for an SKR will be addressed in the context of FD systems.

We are now in a position to present a parameterisation of the nonlinear residual generators.

Let $\Sigma_Q^{\varsigma_0} : \mathcal{Z}^s \rightarrow \mathcal{R}^s$, $\Sigma_Q^{\varsigma_0} \neq 0$, be a stable system operator that satisfies $\Sigma_Q^{\varsigma_0}(0) = 0$. Consider the cascade connection $\Sigma_Q^{\varsigma_0} \circ K_{\Sigma}^{\hat{x}_0}$. Since in the fault-free case for $\hat{x}_0 = x_0$,

$$z = K_{\Sigma}^{\hat{x}_0} \begin{pmatrix} u \\ y \end{pmatrix} = 0,$$

we have

$$\Sigma_Q^{\varsigma_0} \circ K_{\Sigma}^{\hat{x}_0} \begin{pmatrix} u \\ y \end{pmatrix} = \Sigma_Q^{\varsigma_0}(0) = 0.$$

On the other hand, for a detectable fault w, $y = \Sigma_f^{x_0}(u, w) \neq \Sigma^{x_0}(u)$, which leads to

$$z = K_{\Sigma}^{\hat{x}_0} \begin{pmatrix} u \\ y \end{pmatrix} \neq K_{\Sigma}^{\hat{x}_0} \begin{pmatrix} u \\ \Sigma^{x_0}(u) \end{pmatrix} = 0 \Longrightarrow$$
$$\Sigma_Q^{\varsigma_0} \circ K_{\Sigma}^{\hat{x}_0} \begin{pmatrix} u \\ y \end{pmatrix} = \Sigma_Q^{\varsigma_0}(z) \neq 0.$$

Thus, according to Definition 11.2, $\Sigma_Q^{\varsigma_0} \circ K_{\Sigma}^{\hat{x}_0}$ builds a residual generator. This result is summarised in the following theorem.

Theorem 11.1 *Let $K_{\Sigma}^{\hat{x}_0}$ be the coprime SKR of Σ^{x_0} and $\Sigma_Q^{\varsigma_0}$ be any stable system. Then,*

$$R_{\Sigma}^{\xi_0} = \Sigma_Q^{\varsigma_0} \circ K_{\Sigma}^{\hat{x}_0} \tag{11.14}$$

is a stable nonlinear residual generator with $\Sigma_Q^{\varsigma_0}$ as a post-filter.

Note that in the cascade configuration $\Sigma_Q^{\varsigma_0} \circ K_\Sigma^{\hat{x}_0}$, $K_\Sigma^{\hat{x}_0}$ is determined by the system Σ^{x_0} under consideration. In against, the post-filter $\Sigma_Q^{\varsigma_0}$ is a stable system and can be arbitrarily constructed. In this sense, $\Sigma_Q^{\varsigma_0}$ is understood as a parameter operator (system) and the cascade configuration is called parameterisation form of nonlinear residual generators.

In some applications, for instance in a closed-loop feedback control system, the input vector u is a function of the system state variables or output vector. As a result, the SKR of the system can be simply written as

$$K_\Sigma^{\hat{x}_0}(y) = z, z \in \mathcal{R}^m. \tag{11.15}$$

In this case, we have the following theorem, which provides us with a parameterisation of all stable generators in terms of the system SKR.

Theorem 11.2 *Let $K_\Sigma^{\hat{x}_0}$ be the coprime SKR given in (11.15) and $\Sigma_Q^{\varsigma_0}$ be a stable post-filter. Then, any stable residual generator $R_\Sigma^{\xi_0}$ can be parameterised by*

$$R_\Sigma^{\xi_0} = \Sigma_Q^{\varsigma_0} \circ K_\Sigma^{\hat{x}_0}. \tag{11.16}$$

Proof Since $K_\Sigma^{\hat{x}_0}$ is the coprime kernel, we have stable K_Σ^- and (11.12) holds. It follows from the definition of SKR that

$$\begin{aligned} z &= K_\Sigma^{\hat{x}_0} \circ K_\Sigma^-(z) = K_\Sigma^{\hat{x}_0}(y) \Longrightarrow y = K_\Sigma^-(z) \\ &\Longrightarrow R_\Sigma^{\xi_0}(y) = R_\Sigma^{\xi_0} \circ K_\Sigma^-(z) = R_\Sigma^{\xi_0} \circ K_\Sigma^- \circ K_\Sigma^{\hat{x}_0}(y). \end{aligned}$$

Setting

$$\Sigma_Q^{\varsigma_0} = R_\Sigma^{\xi_0} \circ K_\Sigma^- \tag{11.17}$$

gives the final result

$$R_\Sigma^{\xi_0} = \Sigma_Q^{\varsigma_0} \circ K_\Sigma^{\hat{x}_0}.$$

Thus, the theorem is proved.

11.3 Parameterisation of Nonlinear Fault Detection Systems

In the last section, we have discussed about the configuration of the parameterised residual generators. We now extend this work to an overall FD system including a residual generator, an evaluation function and a threshold. Our major focus is on the state space realisation and the existence conditions of the parameterisation of observer-based FD systems. We will first describe the state space configuration, provide the existence conditions and finally characterise the threshold settings corresponding to the two types of evaluation functions, $J_E(r)$ and $J_2(r)$. In this context, nonlinear observer-based FD systems will be parameterised.

11.3.1 State Space Configuration

Consider nonlinear systems (11.1) and the parameterisation form of nonlinear residual generators given in (11.14). We assume that

$$\|h\left(x_1, u\right) - h\left(x_2, u\right)\| \leq \alpha\left(\|x_1 - x_2\|\right)$$

with $\alpha(\cdot)$ being a $\mathcal{K}$-function. Suppose that the state space representation of the SKR $K_{\Sigma}^{\hat{x}_0}$ is of the following form

$$K_{\Sigma}^{\hat{x}_0} : \begin{cases} \dot{\hat{x}} = \phi(\hat{x}, u, y), \hat{x}(0) = \hat{x}_0, \\ z = \varphi(\hat{x}, u, y), \end{cases} \tag{11.18}$$

where $\hat{x} \in \mathcal{R}^n$, $z \in \mathcal{R}^m$. Recall that

$$\hat{y} = y - \varphi(\hat{x}, u, y) \tag{11.19}$$

delivers an estimate for y. Since for every initial condition $\hat{x}(0) = x(0)$, we have

$$\varphi(\hat{x}, u, y) = 0 \Longrightarrow \hat{y} = h\left(\hat{x}, u\right).$$

Thus, it is reasonable to write z as

$$z = y - \hat{y} = y - h\left(\hat{x}, u\right). \tag{11.20}$$

Note that

$$\dot{\hat{x}} = \phi(\hat{x}, u, y), \hat{y} = h\left(\hat{x}, u\right) \tag{11.21}$$

is an output observer. We call system (11.18) with

$$\varphi(\hat{x}, u, y) = y - \hat{y} = y - h\left(\hat{x}, u\right)$$

nonlinear FDF.

Let the state space form of $\Sigma_Q^{x_{q,0}}$ be

$$\Sigma_Q^{x_{q,0}} : \begin{cases} \dot{x}_q = f_q(x_q, z), x_q(0) = x_{q,0}, \\ r = h_q(x_q). \end{cases} \tag{11.22}$$

Then, the state space representation of the parameterised form of the observer-based residual generators is given by

$$R_{\Sigma}^{\xi_0} : \begin{cases} \dot{\hat{x}} = \phi(\hat{x}, u, y), \hat{x}(0) = \hat{x}_0, \\ z = y - \hat{y} = y - h\left(\hat{x}, u\right), \\ \dot{x}_q = f_q(x_q, z), x_q(0) = x_{q,0}, \\ r = h_q(x_q), \end{cases} \quad \xi_0 = \begin{bmatrix} \hat{x}_0 \\ x_{q,0} \end{bmatrix}. \tag{11.23}$$

Recall that two types of evaluation functions $J_E(r)$, $J_2(r)$ and, associated with them, the thresholds $J_{th,E}$, $J_{th,2}$ have been defined, as given in (11.6). Corresponding to them and as defined in Definition 10.2, we call

- an FD system with the observer-based residual generator (11.23), residual evaluation function $J_E(r)$ and threshold $J_{th,E}$ $\mathcal{L}_\infty$-class FD system,
- an FD system with the observer-based residual generator (11.23), residual evaluation function $J_2(r)$ and threshold $J_{th,2}$ $\mathcal{L}_2$-class FD system.

As mentioned in the previous section, $\hat{x}_0$ is in general different from x_0. Consequently, the output of the kernel system is different from zero and depends on $x(0) - \hat{x}(0) = x_0 - \hat{x}_0$ as well as on input u. For the FD purpose, a residual evaluation function and, associated with it, a threshold are needed to avoid false alarms. This is in fact the motivation for our subsequent study on the parameterisation of the threshold settings based on the estimation of the possible influence of $x(0) - \hat{x}(0)$ and u on the residual vector. In the next two subsections, we are going to address the parameterisation of the threshold settings for the $\mathcal{L}_\infty$-class and $\mathcal{L}_2$-class FD systems, respectively.

11.3.2 $\mathcal{L}_\infty$-class FD systems

For our purpose, we first recall a definition which is well known in system stability analysis and serves the characterisation of the $\mathcal{L}_\infty$-class FD systems.

Definition 11.5 *A nonlinear system*

$$\Sigma_Q^{x_{q,0}} : \begin{cases} \dot{x}_q = f_q(x_q, z), x_q(0) = x_{q,0}, \\ r = h_q(x_q) \end{cases} \tag{11.24}$$

is said to be input-to-output stable (IOS) if there exist functions $\beta(\cdot, t) \in \mathcal{KL}$ and $\sigma(\cdot) \in \mathcal{K}$ such that

$$||r(t)|| \leq \beta(||x_{q,0}||, t) + \sigma(||z||_\infty), \ t \geq 0. \tag{11.25}$$

For our purpose, we suppose that system (11.1) is output re-constructible (ORC), as defined in Chap. 10. That is, there exists a nonlinear system

$$\begin{cases} \dot{\hat{x}} = \phi(\hat{x}, u, y), \hat{x}(0) = \hat{x}_0, \\ \hat{y} = h(\hat{x}, u) \end{cases} \tag{11.26}$$

such that $\forall x, \hat{x} \in \mathcal{B}_\delta$, $\|u\|_\infty \leq \delta_u$,

$$||y(t) - \hat{y}(t)|| = ||z(t)|| \leq \beta(||x_0 - \hat{x}_0||, t), \tag{11.27}$$

where $\delta, \delta_u > 0$, $\beta(\cdot, t) \in \mathcal{KL}$.

The following results follow immediately from Definition 11.5 and ORC.

Theorem 11.3 *Assume that system (11.1) is ORC and the post-filter (11.22) is IOS. Let* $x_q(0) = x_{q,0} = 0$*. Then, there exists* $\gamma(\cdot) \in \mathcal{K}$ *so that* $\forall t \geq 0$

$$||r(t)|| \leq \gamma\left(||x_0 - \hat{x}_0||\right) \Longrightarrow J_{th,E} = (\gamma(\delta_o))^2, \tag{11.28}$$

where $\delta_o = \max ||x_0 - \hat{x}_0||$.

Proof It follows from the IOS definition that for $x_q(0) = x_{q,0} = 0, t \geq 0$,

$$\exists \sigma(\cdot) \in \mathcal{K} \text{ s.t. } ||r(t)|| \leq \sigma(||z||_\infty).$$

Furthermore, the output re-constructability of the system (11.1) ensures the existence of a $\mathcal{KL}$-function $\beta(||x_0 - \hat{x}_0||, t)$ so that

$$\begin{aligned} ||z(t)|| \leq \beta(||x_0 - \hat{x}_0||, t) &\leq \max_{||x(0) - \hat{x}(0)|| \leq \delta_o} \beta(||x_0 - \hat{x}_0||, t) \\ &= \beta(\delta_o, 0) \geq ||z||_\infty. \end{aligned}$$

Let

$$\gamma(\delta_o) = \sigma(\beta(\delta_o, 0)).$$

It turns out $\forall t \geq 0$

$$J_E(r) = ||r(t)||^2 \leq (\gamma(\delta_o))^2 =: J_{th,E}.$$

The theorem is thus proved.

Theorem 11.3 reveals that under certain conditions the threshold can be parameterised by the parameter function γ and δ_o, as shown in (11.28). Moreover, applying the existing results on the existence conditions of IOS and ORC to Theorem 11.3 leads to the following corollary.

Corollary 11.1 *Given the system (11.1), the post-filter (11.22) and suppose that there exist*

- *a function $\phi : \mathcal{R}^n \times \mathcal{R}^p \times \mathcal{R}^m \to \mathcal{R}^n$,*
- *functions $V(x, \hat{x}) : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}_+, \varphi_i(\cdot) \in \mathcal{K}, i = 1, 2, 3,$ and constants $\delta, \delta_u > 0$ such that $\forall x, \hat{x} \in \mathcal{B}_\delta, \|u\|_\infty \le \delta_u$,*

$$\varphi_1\left(\left\|y - \hat{y}\right\|\right) \le V(x, \hat{x}) \le \varphi_2\left(\left\|x - \hat{x}\right\|\right),$$
$$V_x(x, \hat{x}) f(x, u) + V_{\hat{x}}(x, \hat{x})\phi(\hat{x}, u, y) \le -\varphi_3\left(\left\|x - \hat{x}\right\|\right),$$

- *$V_q(x_q) : \mathcal{R}^{n_q} \to \mathcal{R}_+, \alpha_1(\cdot), \alpha_2(\cdot) \in \mathcal{K}_\infty$ as well as $\chi(\cdot) \in \mathcal{K}, \alpha_3(\cdot) \in \mathcal{KL}$ such that*

$$\alpha_1(\left\|h(x_q)\right\|) \le V_q(x_q) \le \alpha_2(\left\|x_q\right\|), \forall x_q \in \mathcal{R}^{n_q},$$
$$V_q(x_q) \ge \chi(\|z\|) \Longrightarrow$$
$$\frac{\partial V_q(x_q)}{\partial x_q} f_q(x_q, z) \le -\alpha_3\left(V_q(x_q), \left\|x_q\right\|\right).$$

Then, there exists $\gamma(\cdot) \in \mathcal{K}$ so that $\forall t \ge 0$

$$||r(t)|| \le \gamma\left(||x_0 - \hat{x}_0||\right). \tag{11.29}$$

Proof It follows from Theorem 10.1 that the first two conditions are sufficient for the system under consideration being ORC. Moreover, the third condition is well-known as a necessary and sufficient condition for a system being IOS. As a result of Theorem 11.3, we finally have (11.29).

It is of interest to note that it follows from (11.10) that for $x_0 = \hat{x}_0$ it holds $y = \hat{y}$. Thus, if system (11.1) is ORC, there exists an SKR. Conditions 1 and 2 given in the corollary are sufficient conditions for the existence of the SKR.

11.3.3 $\mathcal{L}_2$-class FD systems

In the sequel, we study $\mathcal{L}_2$-class FD systems. To this end, we first review the definition of $\mathcal{L}_2$-stable systems known in the literature.

Definition 11.6 *A nonlinear system*

$$\Sigma_Q^{x_{q,0}} : \begin{cases} \dot{x}_q = f_q(x_q, z), x_q(0) = x_{q,0}, \\ r = h_q(x_q) \end{cases}$$

is said to be $\mathcal{L}_2$-stable if for some constant $\gamma \ge 0$

$$\|r\|_{2,\tau}^2 \leq \gamma^2 \|z\|_{2,\tau}^2 + \gamma_o(x_{q,0}), \tag{11.30}$$

where $\gamma_o \geq 0$ is a (finite) constant for given $x_{q,0}$.

For the existence of $\mathcal{L}_2$-class FD systems with the parameterisation configuration given in Theorem 11.1 we have the following result.

Theorem 11.4 *Given the observer-based residual generator (11.23) and assume that the post-filter (11.22) is $\mathcal{L}_2$-stable with $x_q(0) = 0$. If there exist*

- *a function $\phi : \mathcal{R}^n \times \mathcal{R}^p \times \mathcal{R}^m \to \mathcal{R}^n$,*
- *functions $V(x, \hat{x}) : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}^+$, $\varphi_1(\cdot) \in \mathcal{K}$, $\varphi_2(\cdot) \in \mathcal{K}_\infty$, such that*

$$0 \leq V(x, \hat{x}) \leq \varphi_1\left(\|x - \hat{x}\|\right),$$

and

$$\begin{aligned} &V_x(x, \hat{x}) f(x, u) + V_{\hat{x}}(x, \hat{x})\phi(\hat{x}, u, y) \\ &\leq -\|z\|^2 + \varphi_2(\|u\|), z = y - h(\hat{x}, u), \end{aligned} \tag{11.31}$$

then it holds

$$\begin{aligned} \|r\|_{2,\tau}^2 &\leq \gamma^2 \int_0^\tau \varphi_2(||u(t)||)dt + \gamma_0 \Longrightarrow \\ J_{th,2} &= \gamma^2 \int_0^\tau \varphi_2(||u(t)||)dt + \gamma_0, \\ \gamma_0 &= \gamma^2 \max_{x_0, \hat{x}_0} \varphi_1\left(\|x_0 - \hat{x}_0\|\right). \end{aligned} \tag{11.32}$$

Proof It follows from (11.31) that

$$\int_0^\tau \|z\|^2 dt \leq \int_0^\tau \varphi_2(||u||)dt + V(x(0), \hat{x}(0)).$$

Since the post-filter (11.22) is $\mathcal{L}_2$-stable with $x_q(0) = 0$ and

$$V(x(0), \hat{x}(0)) \leq \varphi_1\left(\|x_0 - \hat{x}_0\|\right),$$

it turns out

$$\begin{aligned} J_2(r) = \|r\|_{2,\tau}^2 &\leq \gamma^2 \|z\|_{2,\tau}^2 \\ &\leq \gamma^2 \left(\int_0^\tau \varphi_2(||u(t)||)dt + \varphi_1\left(\|x_0 - \hat{x}_0\|\right)\right) \\ &\leq \gamma^2 \int_0^\tau \varphi_2(||u(t)||)dt + \gamma^2 \max_{x_0, \hat{x}_0} \varphi_1\left(\|x_0 - \hat{x}_0\|\right). \end{aligned}$$

As a result, we set

$$J_{th,2} = \gamma^2 \int_0^\tau \varphi_2(||u(t)||)dt + \gamma_0,$$

which completes the proof.

In the FD research, the threshold (11.32) is called adaptive threshold, since it is a function of $||u(t)||$. It is evident that $J_{th,2}$ is parameterised by γ, the $\mathcal{L}_2$-gain of the post-filter, $\varphi_2(||u(t)||)$ and γ_0. It should be pointed out that in the existing studies, an adaptive threshold is generally parameterised by the $\mathcal{L}_2$-gain of the residual generator. In the above study, additional degree of design freedom is introduced in terms of the parameter function φ_2, which can be, for instance, used for the purpose of improving FD performance.

It can be seen that the threshold settings for $J_{th,E}$, $J_{th,2}$ depend on the boundedness of the uncertain initial values $x_0 - \hat{x}_0$. In order to cover all possible situations, this boundedness should be theoretically set very large, which may lead to conservative threshold setting. One way to solve this problem is to apply the so-called randomised algorithms, which provides a tool to handle the uncertainty in the statistical framework and thus lead to an efficient threshold setting.

11.4 Notes and References

In this chapter, we have addressed parameterisation of nonlinear observer-based FD systems. Motivated by the known parameterisation scheme for LTI residual generators and the important role of a parameterisation in FD system analysis and optimisation [1], we have studied the parameterisation issues in two steps. With the aid of nonlinear factorisation and input-output operator theories, it has been first proved that any stable residual generator can be parameterised by a cascade connection of the process SKR and a post-filter that represents the parameter system. In the second step, based on the state space representation of the parameterised residual generator, $\mathcal{L}_\infty$- and $\mathcal{L}_2$-classes FD systems have been investigated. As a result, the threshold settings for both classes of FD systems have been parameterised and, associated with them, some existence conditions have been characterised.

Nonlinear factorisation and input-output operator theories are mathematical methods that have been intensively applied to the investigation of parameterisation of nonlinear stabilising controllers in the 90s. We have referred a series of publications on this topic [2–5] for our work on the parameterisation of nonlinear residual generations. In our study on $\mathcal{L}_\infty$-class observer-based FD systems, the concept of input-to-output stability described in Definition 11.5 and some relevant results play an essential role. We refer the reader to [6, 7] for details. Concerning the $\mathcal{L}_2$-class FD systems, the definition of $\mathcal{L}_2$-stable systems, Definition 11.6, has been adopted, which can be found in [8].

In real applications, disturbances can considerably affect the FD performance. In order to deal with nonlinear systems with unknown inputs like disturbances, most of the results achieved in this chapter should be extended to take into account the disturbances.

It is obvious that applying the developed approaches to the design of $\mathcal{L}_\infty$- and $\mathcal{L}_2$-classes FD systems is a challenging task. It deals with solving those inequalities as given in Corollary 11.1 or in Theorem 11.4. A possible solution for them is to apply some well-established nonlinear techniques. The first efforts to this end have been reported in [9–11], in which the Takagi-Sugeno (T-S) fuzzy technique has been successfully applied to the design of nonlinear observer-based FD systems.

References

1. S. X. Ding, *Model-Based Fault Diagnosis Techniques—Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.
2. A. Paice and A. Van der Schaft, “Stable kernel representations as nonlinear left coprime factorizations,” *Proc. of the 33rd IEEE CDC*, pp. 2786–2791, 1994.
3. W.-M. Lu, “A state-space approach to parameterization of stabilizing controllers for nonlinear systems,” *IEEE Trans. on Automatic Control*, vol. 40, pp. 1576–1588, 1995.
4. A. Paice and A. Van der Schaft, “The class of stabilizing nonlinear plant controller pairs,” *IEEE Trans. on Automat. Control*, vol. 41, pp. 634–645, 1994.
5. K. Fujimoto and T. Sugie, “State-space characterization of Youla parametrization for nonlinear systems based on input-to-state stability,” *Proc. of the 37th IEEE CDC*, vol. 4, pp. 2479–2484, 1998.
6. E. Sontag and Y. Wang, “Notations of input to output stability,” *Syst. Control Lett.*, vol. 38, pp. 235–248, 1999.
7. E. Sontag and Y. Wang, “Characterization of input to output stability,” *SIAM. J. Control Optim.*, vol. 39, pp. 226–249, 2001.
8. A. Van der Schaft, *L2—Gain and Passivity Techniques in Nonlinear Control*. London: Springer, 2000.
9. L. Li, S. X. Ding, J. Qui, Y. Yang, and Y. Zhang, “Weighted fuzzy observer-based fault detection approach for discrete-time nonlinear systems via piecewise-fuzzy lyapunov functions,“ *IEEE Trans. on Fuzzy Systems*, vol. 24, pp. 1320–1333, 2016.
10. L. Li, S. X. Ding, J. Qui, Y. Yang, and D. Xu, “Fuzzy observer-based fault detection design approach for nonlinear processes,” *IEEE Trans. on Syst., Man, and Cybernetics: Systems*, vol. 47, pp. 1941–1952, 2017.
11. L. Li, S. X. Ding, J. Qiu, and Y. Yang, “Real-time fault detection approach for nonlinear systems and its asynchronous T-S fuzzy observer-based implementation,” *IEEE Trans. on Cybernetics*, vol. 47, pp. 283–294, 2017.

Chapter 12
Optimal Fault Detection of a Class of Nonlinear Systems

Having investigated the existence conditions and parameterisation of nonlinear observer-based fault detection systems in the previous two chapters, we now devote our attention to the solution of the optimal fault detection problem formulated in Definition 2.4, the so-called *FD with maximum fault detectability,* for a class of nonlinear systems.

Recall that Theorem 2.1 provides us with a general form of the solutions for the problem of *FD with maximum fault detectability.* For linear dynamic systems, such a solution can be realised by a co-inner-outer factorisation, as demonstrated in Chaps. 4 and 7. This motivates our study on a co-inner-outer factorisation of a class of nonlinear systems aiming at introducing a tool to deal with nonlinear FD issues in a systematic way.

The basic idea behind this work is sketched in Fig. 12.1, where Σ stands for the system

$$y = \Sigma(d)$$

with the output vector y and unknown input vector d, and

$$\Sigma = \Pi \circ \Theta$$

represents the co-inner-outer factorisation of Σ with Π as a co-outer and Θ co-inner. As a result, an optimal residual generation is realised by implementing

$$r = \Pi^{-1} y = \Pi^{-1} \circ \Pi \circ \Theta(d) = \Theta(d).$$

That is, the inverse of the co-outer Π^{-1} is the post-filter in the configuration of residual generator parametrisation. It allows then an (optimal) threshold setting

$$J_{th} = \delta_d = \sup_d \|d\|_2 \,,$$

and the detection logic

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_12

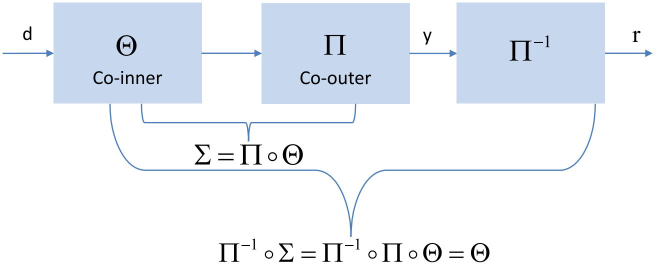


Fig. 12.1 Schematic description of co-inner-outer factorisation based fault detection

$$J = \|r\|_2 \, , J - J_{th} = \begin{cases} \leq 0, & \text{fault-free,} \\ > 0, & \text{faulty.} \end{cases}$$

Our subsequent study will focus on the definition and solution of co-inner-outer factorisation of a class of nonlinear systems. The objective is to introduce a tool to deal with nonlinear FD issues in a systematic way. Our major attention will be paid to the basic ideas, concepts and design schemes.

12.1 System Models and Preliminaries

12.1.1 System Model and Hamiltonian Systems

Consider nonlinear affine systems

$$\Sigma : \dot{x} = a(x) + B(x)d, y = c(x) + D(x)d, \tag{12.1}$$

where $x \in \mathcal{R}^n$, $y \in \mathcal{R}^m$ are process state and output vectors, and $d \in \mathcal{R}^p$ denotes the (unknown) input vector. $a(x)$, $B(x)$, $c(x)$, $D(x)$ are smooth functions of appropriate dimensions. The system model (12.1) can be understood as the dynamic model of an observer-based residual generator with d denoting the composition of the system input and disturbance vectors. Our task, remembering the discussion at the beginning of this chapter, is to design a post-filter. For our purpose, we assume that

- system Σ is stable and $x = 0$ is the asymptotically stable equilibrium point of $a(x)$,
- $p \geq m$ and $D(x)D^T(x)$ is invertible.

The Hamiltonian extension of Σ is a dynamic system described by

$$\begin{aligned}
\dot{x} &= a(x) + B(x)d, \\
\dot{p} &= -\left(\frac{\partial a}{\partial x}(x) + \frac{\partial B}{\partial x}(x)d\right)^T p - \left(\frac{\partial c}{\partial x}(x) + \frac{\partial D}{\partial x}(x)d\right)^T d_a, \\
y &= c(x) + D(x)d, \\
y_a &= B^T(x)p + D^T(x)d_a, \\
y_a &\in \mathcal{R}^p, d_a \in \mathcal{R}^m
\end{aligned}$$

with inputs (d, d_a), outputs (y, y_a) and state variables (x, p) . Let

$$d_a = y = c(x) + D(x)d. \tag{12.2}$$

We then obtain the Hamiltonian system $(D\Sigma)^T \circ \Sigma$ as

$$y_a = (D\Sigma)^T \circ \Sigma\,(d)\,,$$

$$(D\Sigma)^T \circ \Sigma : \begin{cases} \dot{x} = a(x) + B(x)d, \\ \dot{p} = -\left(\frac{\partial a}{\partial x}(x) + \frac{\partial B}{\partial x}(x)d\right)^T p \\ \qquad -\left(\frac{\partial c}{\partial x}(x) + \frac{\partial D}{\partial x}(x)d\right)^T (c(x) + D(x)d)\,, \\ y_a = B^T(x)p + D^T(x)\,(c(x) + D(x)d)\,. \end{cases} \tag{12.3}$$

Analogue to it, defining

$$y_a = d \Longrightarrow d = B^T(x)p + D^T(x)d_a, \tag{12.4}$$

leads to the Hamiltonian system

$$y = \Sigma \circ (D\Sigma)^T\,(d_a)\,,$$

$$\Sigma \circ (D\Sigma)^T : \begin{cases} \dot{x} = a(x) + B(x)B^T(x)p + B(x)D^T(x)d_a, \\ \dot{p} = -\left(\frac{\partial a}{\partial x}(x) + \frac{\partial B}{\partial x}(x)\left(B^T(x)p + D^T(x)d_a\right)\right)^T p \\ \qquad -\left(\frac{\partial c}{\partial x}(x) + \frac{\partial D}{\partial x}(x)\left(B^T(x)p + D^T(x)d_a\right)\right)^T d_a, \\ y = c(x) + D(x)B^T(x)p + D(x)D^T(x)d_a. \end{cases} \tag{12.5}$$

Now, let

$$H\,(x, p, d) = p^T\,(a(x) + B(x)d) + \frac{1}{2}\,(c(x) + D(x)d)^T\,(c(x) + D(x)d)\,, \tag{12.6}$$

$$\begin{aligned} H\,(x, p, d_a) = p^T &\left(a(x) + \frac{1}{2}B(x)B^T(x)p + B(x)D^T(x)d_a\right) \\ &+ c^T(x)d_a + \frac{1}{2}d_a^T D(x)D^T(x)d_a \end{aligned} \tag{12.7}$$

be the Hamiltonian functions, then the Hamiltonian systems $(D\Sigma)^T \circ \Sigma$, $\Sigma \circ (D\Sigma)^T$ can be further written, respectively, as

$$(D\Sigma)^T \circ \Sigma : \begin{cases} \dot{x} = \frac{\partial H}{\partial p}(x, p, d), \\ \dot{p} = -\frac{\partial H}{\partial x}(x, p, d), \\ y_a = \frac{\partial H}{\partial d}(x, p, d), \end{cases} \tag{12.8}$$

$$\Sigma \circ (D\Sigma)^T : \begin{cases} \dot{x} = \frac{\partial H}{\partial p}(x, p, d_a), \\ \dot{p} = -\frac{\partial H}{\partial x}(x, p, d_a), \\ y = \frac{\partial H}{\partial d_a}(x, p, d_a). \end{cases} \tag{12.9}$$

In the literature, it is well-known that for LTI systems, the Hamiltonian systems $(D\Sigma)^T \circ \Sigma$ and $\Sigma \circ (D\Sigma)^T$ are, on the assumption that Σ is expressed by transfer function matrix $G(s)$,

$$(D\Sigma)^T \circ \Sigma = G^T(-s)\, G(s),\ \Sigma \circ (D\Sigma)^T = G(s) G^T(-s),$$

respectively. For our purpose of investigating co-inner-outer as well as inner-outer factorisations, the Hamiltonian systems $(D\Sigma)^T \circ \Sigma$ and $\Sigma \circ (D\Sigma)^T$ are essential.

12.1.2 Inner

Recall that the basic idea behind our work is to construct such a residual generator whose dynamics is co-inner. To this end, we first study the definition and existence conditions of a co-inner system. Unfortunately, there are few existing results on this topic. This motivates us to begin with a review of the definition and associated results on an inner system, which can be found in the literature to some extent.

For an LTI system $y(s) = G(s)d(s)$, it is inner if

$$G^T(-s)\, G(s) = I,$$

which means in turn

$$y_a(s) = d(s),\ y_a(s) = G^T(-s)\, y(s).$$

From this point of view, Scherpen and van der Schaft have, in their work in 1994, introduced the definition that the nonlinear affine system (12.1) is inner if it holds, for the Hamiltonian system $(D\Sigma)^T \circ \Sigma$ (12.8),

$$y_a = d. \tag{12.10}$$

In addition, an important application of an inner system is the property that

$$\|y\|_2 = \|d\|_2 .$$

This property can be equivalently described in the framework of energy balance and is called lossless with respect to the ($\mathcal{L}_2$-gain) supply rate

$$s(d, y) = \frac{1}{2}d^T d - \frac{1}{2}y^T y.$$

That is, there exists a storage function $P(x) \geq 0$, $P(0) = 0$ so that

$$P\left(x\left(t_2\right)\right) - P\left(x\left(t_1\right)\right) = \int_{t_1}^{t_2} s(d, y)d\tau = \frac{1}{2}\int_{t_1}^{t_2} \left(d^T d - y^T y\right) d\tau. \tag{12.11}$$

In their work on inner-outer factorisation, Petersen and van der Schaft have defined an inner system by means of the lossless property of a system. They call the nonlinear affine system (12.1) inner if it is lossless with respect to the above defined $\mathcal{L}_2$-gain supply rate.

It is a well-known result that the existence condition for the nonlinear affine system (12.1) being lossless in sense of (12.11) is that

$$P_x\left(x\right) = \frac{\partial P(x)}{\partial x}$$

solves the following equations

$$P_x\left(x\right) a(x) + \frac{1}{2}c^T(x)c(x) = 0, \tag{12.12}$$

$$P_x\left(x\right) B(x) + c^T(x)D(x) = 0, \tag{12.13}$$

$$D^T(x)D(x) = I. \tag{12.14}$$

Recall that for Hamiltonian system (12.3)

$$y_a = B^T(x)p + D^T(x)\left(c(x) + D(x)d\right).$$

As a result, for $p^T(x) = P_x\left(x\right)$, it holds, according to (12.13) and (12.14),

$$y_a = d.$$

Notice that for $p^T(x) = P_x\left(x\right)$,

$$\begin{aligned} H\left(x, p, d\right) &= p^T\left(a(x) + B(x)d\right) + \frac{1}{2}\left(c(x) + D(x)d\right)^T\left(c(x) + D(x)d\right) \\ &= \dot{P}\left(x\right) + \frac{1}{2}y^T y. \end{aligned}$$

On the other hand, it follows from (12.12)–(12.14) that

$$\begin{aligned}
p^T & \left(a(x)+B(x)d\right)+\frac{1}{2}\left(c(x)+D(x)d\right)^T\left(c(x)+D(x)d\right) \\
&= P_x(x)\,a(x)+\frac{1}{2}c^T(x)c(x)+\left(P_x(x)\,B(x)+c^T(x)D(x)\right)d+\frac{1}{2}d^Td \\
&= \frac{1}{2}d^Td.
\end{aligned}$$

Thus, solving (12.12)–(12.14) leads to

$$\dot{P}(x)=\frac{1}{2}d^Td-\frac{1}{2}y^Ty.$$

In summary, we can claim that both definitions for the nonlinear affine system (12.1) being inner, (12.10) and (12.11), are equivalent, when there exists $P_x(x)$ that solves (12.12)–(12.14).

12.2 Definition of Co-Inner

To our best knowledge, there exist rarely reported results on the topic of co-inner systems. In their work in 1994, Scherpen and van der Schaft have, analogue to the definition of inner systems, defined a co-inner system as follows: the system (12.1) is called co-inner if it holds for the Hamiltonian system (12.9)

$$y=d_a. \tag{12.15}$$

Moreover, it has been proved that (12.15) becomes true when there exists $P(x)$ that solves

$$c(x)+D(x)B^T(x)P_x^T(x)=0, \tag{12.16}$$

$$D(x)D^T(x)=I. \tag{12.17}$$

In addition, when $P(x)$ also solves the following equation

$$P_x(x)\,a(x)+\frac{1}{2}P_x(x)\,B(x)B^T(x)P_x^T(x)=0, \tag{12.18}$$

it holds, for $p^T=P_x(x)$,

$$\begin{aligned} H\left(x,p,d_a\right) &= p^T\left(a(x)+\frac{1}{2}B(x)B^T(x)p+B(x)D^T(x)d_a\right) \\ &\quad +c^T(x)d_a+\frac{1}{2}d_a^T D(x)D^T(x)d_a \\ &= \frac{1}{2}d_a^T d_a = \frac{1}{2}y^T y. \end{aligned}$$

On the other hand, it can be noticed that

$$\begin{aligned} \dot{P}\left(x\right)+\frac{1}{2}y_a^T y_a &= \dot{P}\left(x\right)+\frac{1}{2}d^T d \\ &= P_x\left(x\right)\left(a(x)+B(x)B^T(x)P_x^T\left(x\right)+B(x)D^T(x)d_a\right) \\ &\quad +\frac{1}{2}\left(B^T(x)P_x^T\left(x\right)+D^T(x)d_a\right)^T\left(B^T(x)P_x^T\left(x\right)+D^T(x)d_a\right), \end{aligned}$$

which is obviously different from $H\left(x,p,d_a\right)$. That means, in other words, a co-inner system defined above is not lossless with respect to the supply rate

$$\frac{1}{2}d_a^T d_a-\frac{1}{2}y_a^T y_a=\frac{1}{2}y^T y-\frac{1}{2}d^T d.$$

This observation motivates us to introduce a definition of co-inner, which is of lossless property, as an analogue form of the inner definition given by Petersen and van der Schaft.

Notice that for $p^T=P_x\left(x\right)$,

$$H\left(x,p,d_a\right)-y^T d_a=\dot{P}\left(x\right)-\frac{1}{2}y_a^T y_a.$$

In addition, if (12.16)–(12.18) hold, it turns out

$$H\left(x,p,d_a\right)-y^T d_a=-\frac{1}{2}d_a^T d_a,\, y=d_a.$$

As a result, we have

$$\dot{P}\left(x\right)-\frac{1}{2}y_a^T y_a=-\frac{1}{2}d_a^T d_a \Longrightarrow \dot{P}\left(x\right)=\frac{1}{2}y_a^T y_a-\frac{1}{2}d_a^T d_a=\frac{1}{2}d^T d-\frac{1}{2}y^T y.$$

It is known in the literature that

$$H^{\times}\left(x,p,y\right)=H\left(x,p,d_a\right)-y^T d_a \tag{12.19}$$

can be interpreted as the Hamiltonian function of the inverse of $\Sigma\circ\left(D\Sigma\right)^T$,

$$d_a=\left(\Sigma\circ\left(D\Sigma\right)^T\right)^{-1}\left(y\right). \tag{12.20}$$

Considering that

$$\begin{gathered}
y = c(x) + D(x)B^T(x)p + D(x)D^T(x)d_a \Longrightarrow d_a = E^{-1}(x)\left(y - \hat{c}(x,p)\right), \\
\Longrightarrow \dot{x} = a(x) + B(x)B^T(x)p + B(x)D^T(x)E^{-1}(x)\left(y - \hat{c}(x,p)\right), \\
\hat{c}(x,p) = c(x) + D(x)B^T(x)p,\ E(x) = D(x)D^T(x),\ y_a = d,
\end{gathered}$$

system $\left(\Sigma \circ (D\Sigma)^T\right)^{-1}$ can be written as

$$\left(\Sigma \circ (D\Sigma)^T\right)^{-1} : \begin{cases} \dot{x} = \frac{\partial H^{\times}(x,p,y)}{\partial p}, \\ \dot{p} = -\frac{\partial H^{\times}(x,p,y)}{\partial x}, \\ d_a = -\frac{\partial H^{\times}(x,p,y)}{\partial y}, \end{cases} \tag{12.21}$$

where

$$\begin{aligned}
H^{\times}(x,p,y) &= H(x,p,d_a) - y^T d_a \\
&= p^T\left(a(x) + \frac{1}{2}B(x)B^T(x)p + B(x)D^T(x)d_a\right) \\
&\quad + c^T(x)d_a + \frac{1}{2}d_a^T D(x)D^T(x)d_a - y^T d_a \\
&= p^T\left(a(x) + \frac{1}{2}B(x)B^T(x)p\right) - \frac{1}{2}d_a^T D(x)D^T(x)d_a \\
&= p^T\left(a(x) + \frac{1}{2}B(x)B^T(x)p\right) - \frac{1}{2}\left(y - \hat{c}(x,p)\right)^T E^{-1}(x)\left(y - \hat{c}(x,p)\right).
\end{aligned} \tag{12.22}$$

Motivated by the above discussion, we now introduce the definition of a co-inner system.

Definition 12.1 *System (12.1) is called co-inner, when*

- *the input to output map of the Hamiltonian system (12.21) from y to d_a is identity,*
- *there exists a function $P(x) \geq 0$, $P(0) = 0$ such that for all $t_1 \geq t_0$, d*

$$P(x(t_1)) - P(x(t_0)) = \frac{1}{2}\int_{t_0}^{t_1}\left(d^T d - y^T y\right)d\tau. \tag{12.23}$$

We would like to remember that in Sect. 7.4 we have introduced the co-inner definition for linear (time-varying) systems, and discussed about the lossless property of a co-inner system. From the viewpoint that fault detection is indeed an information extraction problem, we have introduced the concept lossless with respect to information transform rate. According to this definition, we also call

$$s(d, y) := \frac{1}{2}d^T d - \frac{1}{2}y^T y$$

information transform rate.

For the existence of co-inner system, we have the following theorem.

Theorem 12.1 *System (12.1) is co-inner, when*

$$\frac{\partial P(x)}{\partial x}a(x)+\frac{1}{2}\frac{\partial P(x)}{\partial x}B(x)B^T(x)\frac{\partial P^T(x)}{\partial x} = 0, \tag{12.24}$$

$$c(x)+D(x)B^T(x)\frac{\partial P^T(x)}{\partial x} = 0, \tag{12.25}$$

$$D(x)D^T(x) = I \tag{12.26}$$

are solvable for $P(x) \geq 0$.

The proof of the above theorem is straightforward along the lines given in the literature for the proof of the existence conditions of an inner system. See also the above discussion.

12.3 Co-inner-outer Factorisation

12.3.1 Basic Idea

With the definition of co-inner, we are now in a position to begin with the study on our initial problem: find a solution for the co-inner-outer factorisation problem. To this end, we follow the idea of an existing solution for the inner-outer factorisation problem, which has been reported in the paper by Ball and van der Schaft in 1996.

Suppose that

$$\Sigma = \Pi \circ \Theta$$

is a co-inner-outer factorisation of the nonlinear affine system Σ given in (12.1) with co-inner Θ and co-outer Π. We denote the state space realisation of Π by

$$\Pi : \begin{cases} \dot{\bar{x}} = \bar{a}(\bar{x}) + \bar{B}(\bar{x})\bar{y}, \\ y = \bar{c}(\bar{x}) + \bar{D}(\bar{x})\bar{y}, \end{cases} \tag{12.27}$$

where $\bar{y}$ is the output of the co-inner Θ. Moreover, let

$$H^{\times}(\bar{x}, \bar{p}, y) = \bar{p}^T \left(\bar{a}(\bar{x}) + \frac{1}{2}\bar{B}(\bar{x})\bar{B}^T(\bar{x})\bar{p}\right)$$
$$- \frac{1}{2}\left(y - \hat{c}(\bar{x}, \bar{p})\right)^T \bar{E}^{-1}(\bar{x})\left(y - \hat{c}(\bar{x}, \bar{p})\right)$$

be Hamiltonian function of $\left(\Pi \circ (D\Pi)^T\right)^{-1}$ with

$$\bar{E}(\bar{x}) = \bar{D}(\bar{x})\bar{D}^T(\bar{x}), \hat{c}(\bar{x}, \bar{p}) = \bar{c}(\bar{x}) + \bar{D}(\bar{x})\bar{B}^T(\bar{x})\bar{p}.$$

We have

$$\left(\Pi \circ (D\Pi)^T\right)^{-1} : \begin{cases} \dot{\bar{x}} = \frac{\partial H^\times(\bar{x},\bar{p},y)}{\partial \bar{p}}, \\ \dot{\bar{p}} = -\frac{\partial H^\times(\bar{x},\bar{p},y)}{\partial \bar{x}}, \\ \bar{d}_a = \bar{E}^{-1}(\bar{x})\left(y - \hat{c}(\bar{x}, \bar{p})\right) = -\frac{\partial H^\times(\bar{x},\bar{p},y)}{\partial y}. \end{cases} \tag{12.28}$$

Since Θ is co-inner, it holds

$$\Sigma \circ (D\Sigma)^T = \Pi \circ (D\Pi)^T \Longrightarrow \left(\Sigma \circ (D\Sigma)^T\right)^{-1} = \left(\Pi \circ (D\Pi)^T\right)^{-1}. \tag{12.29}$$

It follows from the pioneering work by Ball and van der Schaft that there should exist a canonical transformation, $(x, p) \longrightarrow (\bar{x}, \bar{p})$, such that

$$H^\times(\bar{x}, \bar{p}, y) = H^\times(x, p, y). \tag{12.30}$$

Analogue to the procedure proposed by Ball and van der Schaft for finding an outer system, in the subsequent work we will solve the co-inner-outer factorisation problem by

- firstly determining the canonical transformation $(x, p) \longrightarrow (\bar{x}, \bar{p})$ which results in (12.30),
- based on it, finding $\bar{a}(\bar{x}), \bar{B}(\bar{x}), \bar{c}(\bar{x}), \bar{D}(\bar{x})$ in the co-outer (12.27), and
- finally checking if
$$\Theta = \Pi^{-1} \circ \Sigma$$
is co-inner.

12.3.2 Solution

We begin with the first step aiming at finding the canonical transformation $(x, p) \longrightarrow (\bar{x}, \bar{p})$. To this end, we apply the so-called generating function approach which is known and well-established in Hamiltonian mechanics. Roughly speaking, the generating function approach consists in finding a generating function that defines a coordinates transformation and guarantees the resulted coordinates transformation being canonical.

There are four types of generating functions. For our purpose, we adopt the type 2 generating function, which should be, for time-invariant systems, a function of $x, \bar{p}$. We denote it by

$$G_2(x, \bar{p}).$$

It is well-known that the following equations,

$$p = \frac{\partial G_2(x, \bar{p})}{\partial x}, \tag{12.31}$$

$$\bar{x} = \frac{\partial G_2(x, \bar{p})}{\partial \bar{p}}, \tag{12.32}$$

define the canonical transformation $(x, p) \longrightarrow (\bar{x}, \bar{p})$. For our problem solution, we define

$$G_2(x, \bar{p}) = P(x) + \bar{p}^T x. \tag{12.33}$$

It yields

$$p = P_x^T(x) + \bar{p}, \, P_x(x) = \frac{\partial P(x)}{\partial x}, \tag{12.34}$$

$$\bar{x} = x. \tag{12.35}$$

Having determined the canonical transformation

$$(x, p) \longrightarrow (\bar{x}, \bar{p}) = (x, p - P_x^T(x)),$$

we start with the second step of determining $\bar{a}(\bar{x}), \bar{B}(\bar{x}), \bar{c}(\bar{x}), \bar{D}(\bar{x})$ based on the equation

$$H^{\times}(\bar{x}, \bar{p}, y) = H^{\times}(x, p, y). \tag{12.36}$$

To this end, substituting $\bar{x}$ in $H^{\times}(\bar{x}, \bar{p}, y)$ by x and p in $H^{\times}(x, p, y)$ by $P_x^T(x) + \bar{p}$, respectively, leads to

$$\begin{aligned}
H^{\times}(\bar{x}, \bar{p}, y) &= \bar{p}^T \left(\bar{a}(x) + \frac{1}{2} \bar{B}(x) \bar{B}^T(x) \bar{p} \right) \\
&\quad - \frac{1}{2} \left(y - \hat{c}(x, \bar{p}) \right)^T \bar{E}^{-1}(x) \left(y - \hat{c}(x, \bar{p}) \right), \\
H^{\times}(x, p, y) &= \left(P_x^T(x) + \bar{p} \right)^T \left(a(x) + \frac{1}{2} B(x) B^T(x) \left(P_x^T(x) + \bar{p} \right) \right) \\
&\quad - \frac{1}{2} \left(y - \hat{c}(x, P_x^T(x) + \bar{p}) \right)^T E^{-1}(x) \left(y - \hat{c}(x, P_x^T(x) + \bar{p}) \right).
\end{aligned}$$

Now, comparing $H^{\times}(\bar{x}, \bar{p}, y)$ and $H^{\times}(x, p, y)$ given above under the condition (12.36) gives

$$\begin{gathered} y^T \bar{E}^{-1}(x) y = y^T E^{-1}(x) y \Longrightarrow \bar{E}(x) = E(x) \Longrightarrow \bar{D}(\bar{x}) = E^{1/2}(x), \\ \hat{c}(x, P_x^T(x) + \bar{p})^T E^{-1}(x) y = \hat{c}(x, \bar{p})^T \bar{E}^{-1}(x) y \Longrightarrow \\ \bar{D}(\bar{x}) \bar{B}^T(\bar{x}) = D(x) B^T(x) \Longrightarrow \bar{B}(\bar{x}) = B(x) D^T(x) E^{-1/2}(x), \\ \bar{c}(\bar{x}) = c(x) + D(x) B^T(x) P_x^T(x). \end{gathered}$$

Moreover, it holds

$$\begin{gathered} \bar{p}^T \left(\bar{a}(x) + \frac{1}{2} \bar{B}(x) \bar{B}^T(x) \bar{p} \right) = \\ \left(\bar{p} + P_x^T(x) \right)^T \left(a(x) + \frac{1}{2} B(x) B^T(x) \left(\bar{p} + P_x^T(x) \right) \right) \Longrightarrow \\ \bar{p}^T \bar{a}(x) = \bar{p}^T \left(a(x) + B(x) B^T(x) P_x^T(x) \right) \\ \Longrightarrow \bar{a}(x) = a(x) + B(x) B^T(x) P_x^T(x), \\ \bar{p}^T \left(B(x) B^T(x) - \bar{B}(x) \bar{B}^T(x) \right) \bar{p} = 0 \Longrightarrow \bar{p} = 0, \\ P_x(x)\, a(x) + \frac{1}{2} P_x(x)\, B(x) B^T(x) P_x^T(x) = 0. \end{gathered} \tag{12.37}$$

Equation (12.37) is the so-called Hamilton-Jacobi-Bellman (HJB) equation with $P_x(x)$ as its solution. As a result, we have the co-outer Π (as defined in (12.27)) described by

$$\Pi : \begin{cases} \dot{x} = a(x) + B(x) B^T(x) P_x^T(x) + B(x) D^T(x) E^{-1/2}(x) \bar{y}, \\ y = c(x) + D(x) B^T(x) P_x^T(x) + E^{1/2}(x) \bar{y}. \end{cases} \tag{12.38}$$

Since the inverse of (12.27) is

$$\Pi^{-1} : \begin{cases} \dot{\bar{x}} = \bar{a}(\bar{x}) + \bar{B}(\bar{x}) \bar{D}^{-1}(\bar{x}) \left(y - \bar{c}(\bar{x}) \right) \\ \quad = \bar{a}(\bar{x}) - \bar{B}(\bar{x}) \bar{D}^{-1}(\bar{x}) \bar{c}(\bar{x}) + \bar{B}(\bar{x}) \bar{D}^{-1}(\bar{x}) y, \\ \bar{y} = \bar{D}^{-1}(\bar{x}) \left(y - \bar{c}(\bar{x}) \right), \end{cases} \tag{12.39}$$

it follows from (12.38) that

$$\Pi^{-1} : \begin{cases} \dot{\bar{x}} = a(\bar{x}) + B(\bar{x}) \left(I - D^T(\bar{x}) E^{-1}(\bar{x}) D(\bar{x}) \right) B^T(\bar{x}) P_x^T(\bar{x}) \\ \qquad + B(\bar{x}) D^T(\bar{x}) E^{-1}(\bar{x}) \left(y - c(\bar{x}) \right), \\ \bar{y} = E^{-1/2}(\bar{x}) y - E^{-1/2}(\bar{x}) \left(c(\bar{x}) + D(\bar{x}) B^T(\bar{x}) P_x^T(\bar{x}) \right). \end{cases} \tag{12.40}$$

Our final step is to check if

$$\Theta = \Pi^{-1} \circ \Sigma$$

is co-inner. To this end, denote the state space model of Θ by

$$\Theta = \Pi^{-1} \circ \Sigma : \begin{cases} \dot{x}_\theta = a_\theta(x_\theta) + B_\theta(x_\theta)d, \\ \bar{y} = c_\theta(x_\theta) + D_\theta(x_\theta)d. \end{cases} \tag{12.41}$$

Recall that for $p = P_x^T(x)$,

$$\begin{aligned} d_a &= E^{-1}(x)\left(y - c(x) - D(x)B^T(x)P_x^T(x)\right) \Longrightarrow \\ d &= B^T(x)P_x^T(x) + D^T(x)d_a \\ &= B^T(x)P_x^T(x) + D^T(x)E^{-1}(x)\left(y - c(x) - D(x)B^T(x)P_x^T(x)\right), \end{aligned} \tag{12.42}$$

which results in

$$\Pi^{-1} \circ \Sigma : \begin{cases} \dot{\bar{x}} = a(\bar{x}) + B(\bar{x})\left(I - D^T(\bar{x})E^{-1}(\bar{x})D(\bar{x})\right)B^T(\bar{x})P_x^T(\bar{x}) \\ \qquad +B(\bar{x})D^T(\bar{x})E^{-1}(\bar{x})\left(y - c(\bar{x})\right), \\ \dot{x} = a(x) + B(x)\left(I - D^T(x)E^{-1}(x)D(x)\right)B^T(x)P_x^T(x) \\ \qquad +B(x)D^T(x)E^{-1}(x)\left(y - c(x)\right), \\ \bar{y} = E^{-1/2}(\bar{x})y - E^{-1/2}(\bar{x})\left(c(\bar{x}) + D(\bar{x})B^T(\bar{x})P_x^T(\bar{x})\right). \end{cases} \tag{12.43}$$

For $x(0) = \bar{x}(0)$, it holds, according to (12.43), $x(t) = \bar{x}(t), t \geq 0$. As a result, $\Theta = \Pi^{-1} \circ \Sigma$ is reduced to

$$\Theta : \begin{cases} \dot{x} = a(x) + B(x)\left(I - D^T(x)E^{-1}(x)D(x)\right)B^T(x)P_x^T(x) \\ \qquad +B(x)D^T(x)E^{-1}(x)\left(y - c(x)\right), \\ \bar{y} = E^{-1/2}(x)y - E^{-1/2}(x)\left(c(x) + D(x)B^T(x)P_x^T(x)\right), \end{cases}$$

which can be further written as, considering (12.42),

$$\Theta : \begin{cases} \dot{x} = a(x) + B(x)d, \\ \bar{y} = -E^{-1/2}(x)D(x)B^T(x)P_x^T(x) + E^{-1/2}(x)D(x)d. \end{cases} \tag{12.44}$$

In other words,

$$\begin{aligned} &a_\theta(x_\theta) = a(x), B_\theta(x_\theta) = B(x), D_\theta(x_\theta) = E^{-1/2}(x)D(x), \\ &c_\theta(x_\theta) = -E^{-1/2}(x)D(x)B^T(x)P_x^T(x). \end{aligned}$$

Note that

$$\begin{aligned} P_x(x)a(x) + \frac{1}{2}P_x(x)B(x)B^T(x)P_x^T(x) &= 0, \\ E^{-1/2}(x)D(x)D^T(x)E^{-1/2}(x) &= I, \\ -E^{-1/2}(x)D(x)B^T(x)P_x^T(x) + E^{-1/2}(x)D(x)B^T(x)P_x^T(x) &= 0. \end{aligned}$$

Therefore, according to Theorem 12.1 Θ, is co-inner.

With co-outer and co-inner given in (12.38) and (12.44) respectively, we have completed a co-inner-outer factorisation.

12.4 Application to Fault Detection

12.4.1 Threshold Setting

An immediate application of the co-inner-outer factorisation is the threshold setting. It follows from (12.23) that, if Π is a co-outer of Σ given in (12.1),

$$r = \Pi^{-1}(y) = \Pi^{-1} \circ \Sigma(d) = \Theta(d)$$

is a co-inner mapping of d to the residual vector r and thus satisfies, in the fault-free case,

$$P(x(\tau)) - P(x(0)) = \frac{1}{2} \int_0^{\tau} \left(d^T d - r^T r \right) dt \Longrightarrow$$
$$\|r\|_{2,\tau}^2 \leq \|d\|_{2,\tau}^2 + 2P(x(0)).$$

When the upper-bounds of $\|d\|_2$, $P(x(0))$ are known and denoted by

$$\sup_d \|d\|_2^2 = \delta_d^2, \sup_{x(0)} P(x(0)) = \delta_o/2,$$

it holds

$$\sup_{d,x(0)} \|r\|_{2,\tau}^2 = \delta_d^2 + \delta_o,$$

and thus the threshold can be defined as

$$J_{th} = \delta_d^2 + \delta_o. \tag{12.45}$$

In real practical applications, residual evaluation often runs with a moving window and the length of the evaluation is set short aiming at an early detection of faults in the system. Correspondingly, we have

$$P(x(\tau_2)) - P(x(\tau_1)) = \frac{1}{2} \int_{\tau_1}^{\tau_2} \left(d^T d - r^T r \right) dt.$$

On the assumption that in the time interval $[\tau_1, \tau_2]$, $P(x(\tau_2)) \approx P(x(\tau_1))$, the threshold can be simply set as

$$J_{th} = \delta_{d,[\tau_1,\tau_2]}^2, \delta_{d,[\tau_1,\tau_2]}^2 = \sup_d \int_{\tau_1}^{\tau_2} \left(d^T d \right) dt,$$

corresponding to the residual evaluation

$$J = \|r\|_{2,[\tau_1,\tau_2]}^2 = \int_{\tau_1}^{\tau_2} \left(r^T r\right) dt.$$

12.4.2 Design of a Post-filter

In Chap. 10, we have studied the design of $\mathcal{L}_2$-NFDF. In this sub-section, we briefly describe a fault detection scheme by adding a post-filter to an $\mathcal{L}_2$-NFDF in order to enhance the fault detectability.

Consider a nonlinear affine system

$$\Sigma : \begin{cases} \dot{x} = a(x) + B(x)u + E(x)d, \\ y = c(x) + D(x)u + F(x)d, \end{cases} \tag{12.46}$$

where $x \in \mathcal{R}^n, u \in \mathcal{R}^{k_u}, y \in \mathcal{R}^m$ are process state, input and output vectors, and $d \in \mathcal{R}^p$ denotes the (unknown) input vector. $a(x)$, $B(x)$, $c(x)$, $D(x)$, $E(x)$ and $F(x)$ are smooth functions of appropriate dimensions. Suppose that an $\mathcal{L}_2$-NFDF of the form

$$\Sigma_{FDF} : \begin{cases} \dot{\hat{x}} = a(\hat{x}) + B(\hat{x})u + L(\hat{x})\left(y - c(\hat{x}) - D(\hat{x})u\right), \\ r_o = y - c(\hat{x}) - D(\hat{x})u, \end{cases}$$

is designed such that

$$\|r_o\|_{2,\tau}^2 \leq \gamma^2 \left\|\bar{d}\right\|_{2,\tau}^2 + \gamma_o, \bar{d} = \begin{bmatrix} u \\ d \end{bmatrix}.$$

Let

$$\bar{x} = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}, f(\bar{x}) = \begin{bmatrix} a(x) \\ a(\hat{x}) + L(\hat{x})\left(c(x) - c(\hat{x})\right) \end{bmatrix},$$
$$G(\bar{x}) = \begin{bmatrix} B(x) & E(x) \\ B(\hat{x}) + L(\hat{x})\left(D(x) - D(\hat{x})\right) & L(\hat{x})F(x) \end{bmatrix},$$
$$h(\bar{x}) = c(x) - c(\hat{x}), \bar{D}(\bar{x}) = \begin{bmatrix} D(x) - D(\hat{x}) & F(x) \end{bmatrix}.$$

The overall system dynamics of Σ_{NFDF} is governed by

$$\Sigma_{NFDF} : \begin{cases} \dot{\bar{x}} = f(\bar{x}) + G(\bar{x})\bar{d}, \\ r_o = h(\bar{x}) + \bar{D}(\bar{x})\bar{d}. \end{cases} \tag{12.47}$$

On the assumption that $m \leq p + k_u$ and $\bar{D}(\bar{x})\bar{D}^T(\bar{x})$ is invertible, we compute co-inner-outer factorisation, as described in the previous section, and denote it by

$$\Sigma_{NFDF} = \Pi_{NFDF} \circ \Theta_{NFDF}.$$

The post-filter and the overall residual generator are given by

$$Q = \Pi_{NFDF}^{-1}, r = Q(r_o) = Q \circ \Sigma_{FDF}(u, y),$$

respectively. Using the residual evaluation function

$$J = \|r\|_{2,[\tau_1,\tau_2]}^2 = \int_{\tau_1}^{\tau_2} \left(r^T r\right) dt,$$

the threshold can be finally set as

$$J_{th} = \|u\|_{2,[\tau_1,\tau_2]}^2 + \delta_{d,[\tau_1,\tau_2]}^2,$$
$$\delta_{d,[\tau_1,\tau_2]}^2 = \sup_d \int_{\tau_1}^{\tau_2} \left(d^T d\right) dt, \ \|u\|_{2,[\tau_1,\tau_2]}^2 = \int_{\tau_1}^{\tau_2} \left(u^T u\right) dt,$$

when in the time interval $[\tau_1, \tau_2]$, $P(x(\tau_2)) \approx P(x(\tau_1))$.

12.5 Notes and References

The major focus of this chapter is on investigating co-inner-outer factorisation of nonlinear affine systems, which results in an optimal solution for the fault detection problem formulated in Definition 2.4.

The topic of inner-outer factorisation of nonlinear affine systems was extensively studied in the 1990s. Unfortunately, very few results were reported on co-inner-outer factorisation of nonlinear systems, although this is a dual problem of the inner-outer factorisation. This observation has motivated and driven us to deal with co-inner-outer factorisation of nonlinear affine systems, even though this is not our original intention for the FD study. Considering that, for the intended FD study, necessary mathematical knowledge is, more or less, out of the scope of this book, efforts have been made to solve the addressed problems as simple as possible, even if they may not be the elegant way of handling.

The so-called Hamiltonian extension of the nonlinear system Σ under consideration was introduced by Crouch and van der Schaft [1] and is essential for dealing with inner-outer factorisation issues in the state space representation. Analogue to the expressions in the linear case,

$$G^T(-s)\,G(s),\, G(s)G^T(-s)\,,$$

the Hamiltonian systems,

$$(D\Sigma)^T \circ \Sigma,\, \Sigma \circ (D\Sigma)^T\,,$$

have been introduced based on the Hamiltonian extension [2, 3]. The introduction of Hamiltonian function allows us not only to express $(D\Sigma)^T \circ \Sigma$, $\Sigma \circ (D\Sigma)^T$ in a compact form, but also to deal with inner-outer factorisation issues in the framework of energy balance. It is remarkable that the definitions of inner introduced in [2] and [3] are slight different, as mentioned in Sect. 12.1.

For our work, the definition of co-inner plays a central role. Regrettably, we have only found the definition introduced in [2] using the input and output relation of $\Sigma \circ (D\Sigma)^T$, instead of a definition in the context of energy balance. This motivates us to adopt the analogue form of the inner definition introduced in [3, 4]. Moreover, we have also noticed the problem of using the Hamiltonian function associated with $\Sigma \circ (D\Sigma)^T$ [2] for the definition of co-inner in the context of energy balance. Inspired by the discussions in [3, 4], we have decided to study the co-inner issues from the aspect of $\left(\Sigma \circ (D\Sigma)^T\right)^{-1}$ with the corresponding Hamiltonian function $H^\times(x, p, y)$ given in (12.22). In fact, considering that $\left(\Sigma \circ (D\Sigma)^T\right)^{-1}$ is driven by y, the system measurement vector, this view of co-inner is consistent with our intention of investigating fault detection issues. On the basis of these careful considerations, we have introduced Definition 12.1 for co-inner.

Along the lines of solving inner-outer factorisation given in [3, 4], we have studied co-inner-outer factorisation. The key step in our solution of co-inner-outer factorisation is to determine the canonical transformation $(x, p) \longrightarrow (\bar{x}, \bar{p})$. A canonical transformation is a change of canonical coordinates, as we have intended with $(x, p) \longrightarrow (\bar{x}, \bar{p})$, that preserves the forms of the Hamiltonian function and system. Different from the differential geometric methods reported in [3, 4], we have applied the generating function approach, a classic technique known in Hamiltonian mechanics [5].

The application of the co-inner-outer factorisation to fault detection is straightforward and similar with the handling of linear systems. It results in optimal fault detectability and an easy setting of threshold. As described in Sect. 12.4, it can be used in combination with an $\mathcal{L}_2$-NFDF.

It should be remarked that

- although an analytical solution has been derived for a co-inner-outer factorisation, solving HJB equation (12.37) is necessary. It is well-known that solutions of HJB equations are a challenging task, which is in fact a key issue in the design of co-inner-outer factorisation aided FD system design;
- in our study, the stability issues of the co-outer as well as its inverse (acting a post-filter) have not been addressed. We refer the reader to the extensive discussions in [3, 4] on this topic.

References

1. P. Crouch and A. V. der Schaft, *Variational and Hamiltonian Control Systems*. Basel: Springer-Verlag, 1987.
2. J. M. A. Scherpen and A. V. der Schaft, “Normalized coprime factorization and balancing for unstable nonlinear systems,” *Int. J. Control*, vol. 60, pp. 1193–1222, 1994.
3. J. A. Ball and A. J. V. der Schaft, “J-inner-outer factorization, J-spectral factorization, and robust control for nonlinear systems,” *IEEE Trans. on Automatic Contr.*, vol. 41, pp. 379–392, 1996.
4. M. A. Petersen and A. Van der Schaft, “Nonsquare spectral factorization for nonlinear control systems,” *IEEE Trans. on Autom. Control*, vol. 50, pp. 286–298, 2005.
5. H. Goldstein, *Classical Mechanics*. Addison-Wesley Pub, 1980.

Part IV
Statistical and Data-driven Fault Diagnosis Methods

Chapter 13
A Critical Review of MVA-based Fault Detection Methods

It cannot be emphasised too much how popular the multivariate analysis (MVA) based methods are in handling fault diagnosis and process monitoring issues, both in academic research and practical application domains. It is the common opinion that statistical MVA techniques are the fundament in the data-driven fault detection framework. The current enthusiasm for statistical and machine learning (ML) as well as for big data has remarkably promoted the application of MVA-based methods to data-driven fault diagnosis and process monitoring. It can be noticed that, in the course of this development, major research focuses are on the application of novel approaches and algorithms known from statistical and machine learning. Few or even no attention has been paid to the original fault detection and diagnosis problems with their distinct statistical background and requirements. This observation motivates a critical review of basic methods in the framework of MVA-based fault detection methods in this chapter.

The objectives of our critical review are

- to stress and correct popular but misleading use of some standard techniques or methods for the FD purpose,
- to pose critical questions on some basic MVA-based FD methods, and
- to motivate development of alternative MVA-based FD methods.

The structure of this chapter is different from the previous ones. In each section, we are going to address one topic in three steps:

- description and analysis of the method or technique or algorithm to be addressed,
- comments, and
- possible alternative solutions.

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_13

13.1 On Projection Technique and Its Use in Fault Detection

13.1.1 Problem Description

In many data-driven methods, projecting or transforming process data from the measurement subspace to another subspace with a reduced dimension is the state of the art. Among these methods, PCA is the most typical example and widely recognised as a standard data-driven fault detection method. For our discussion, we consider PCA as a reference.

The initial idea of the PCA algorithm is to find a lower dimensional subspace that contains the variation in the process data as much as possible. In this way, a dimensionality reduction is achieved, which can then be applied, for instance, for data compression, data visualisation and interpretation. Below, we briefly summarise the PCA algorithm, which has been introduced in Sect. 3.4:

- Center the process data $y_i \in \mathcal{R}^m, i = 1, \cdots, N$,

$$\bar{y}(N) = \frac{1}{N}\sum_{i=1}^{N} y_i, \bar{y}_i = y_i - \bar{y}(N), \tag{13.1}$$

  and form the data matrix

$$Y_N = \left[\, \bar{y}_1 \cdots \bar{y}_N \,\right] \in \mathcal{R}^{m \times N};$$

- Estimate the covariance matrix

$$\hat{\Sigma} = \frac{1}{N-1} Y_N Y_N^T, \tag{13.2}$$

  and do an SVD of $\hat{\Sigma}$

$$\hat{\Sigma} = P \Lambda P^T, \Lambda = diag\left(\sigma_1^2, \cdots, \sigma_m^2\right), \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_m^2; \tag{13.3}$$

- Determine the number of principal components (PCs) l and decompose P, Λ into

$$\Lambda = \begin{bmatrix} \Lambda_{pc} & 0 \\ 0 & \Lambda_{res} \end{bmatrix}, \Lambda_{pc} = diag\left(\sigma_1^2, \cdots, \sigma_l^2\right), \tag{13.4}$$

$$\Lambda_{res} = diag\left(\sigma_{l+1}^2, \cdots, \sigma_m^2\right) \in \mathcal{R}^{(m-l)\times(m-l)}, \sigma_l^2 >> \sigma_{l+1}^2,$$

$$P = \left[\, P_{pc} \; P_{res} \,\right] \in \mathcal{R}^{m \times m}, P_{pc} \in \mathcal{R}^{m \times l}. \tag{13.5}$$

The principal components represented by P_{pc} are the output of the above PCA algorithm. They span the subspace, onto which the process data are “projected”.

Thus, P_{pc} is the solution for the dimensionality reduction. In fact, it is evident that P_{pc}, together with Λ_{pc}, provides the best estimation of $\hat{\Sigma}$ in the sense of minimising

$$\left\| \hat{\Sigma} - P_{pc} \Lambda_{pc} P_{pc}^T \right\|_F^2 = \sum_{i=l+1}^{m} \sigma_i^2,$$

once the dimension of the subspace l is fixed. P_{pc} can also be determined by solving an optimisation problem,

$$\min_{P_{pc}, Z} \sum_{i=1}^{N} \left\| \bar{y}_i - P_{pc} z_i \right\|^2,$$

$$\text{s.t. } P_{pc}^T P_{pc} = I,$$

$$P_{pc} \in \mathcal{R}^{m \times l}, Z = \begin{bmatrix} z_1 \cdots z_N \end{bmatrix}, z_i \in \mathcal{R}^l, i = 1, \cdots, N,$$

which gives a best fitting of the data matrix Y_N by an l (lower) dimensional subspace.

The questions to be discussed are:

- Is it necessary to "project" the process data onto the principal component subspace for the fault detection purpose?
- Does such a "projection" bring added-value for fault detection?

13.1.2 Discussion: The Pros and Cons

Pro The PCA technique is well-established. Based on the projection of the process data onto two subspaces, the principal component subspace and residual subspace, as described in Sect. 3.4, two test statistics, T_{PCA}^2 and SPE can be defined and applied for the detection purpose.

Con According to the Neyman-Pearson Lemma and GLR method, the T^2-test statistic of the form

$$J_{T^2} = \tilde{y}^T \hat{\Sigma}^{-1} \tilde{y}$$

results in the maximal fault detectability for a (given) significance level, when the process data are (nearly) normally distributed, where $\tilde{y}$ is the centred online measurement data. Considering that $\hat{\Sigma}^{-1}$ can be computed by

$$\hat{\Sigma} = P \Lambda P^T \Longrightarrow \hat{\Sigma}^{-1} = P \Lambda^{-1} P^T,$$

where P, Λ are given in (13.4)–(13.5), the projection P can be used for the purpose of computing $\hat{\Sigma}^{-1}$, which yields

$$J_{T^2} = \tilde{y}^T P \Lambda^{-1} P^T \tilde{y} = \hat{y}^T \Lambda^{-1} \hat{y}, \hat{y} = P^T \tilde{y}.$$

Note that the introduction of the projection P does not lead to any improvement of the FD performance. On the other hand, the strong focus on the principal component subspace in the application of dimensionality reduction is often misinterpreted in the FD study, as calling for more attention to the T_{PCA}^2-test statistic,

$$T_{PCA}^2 = \tilde{y}^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T \tilde{y}.$$

As discussed in Sect. 3.4, it is evident that detecting faults in the principal component subspace is much more difficult than detecting faults in the residual subspace, since the uncertainty caused by the noises, represented in form of the covariance matrix, is stronger in the principal component subspace than the one in the residual subspace. Indeed, in case that

$$l << m,$$

a projection of the process data onto the residual subspace would make sense, thanks to the weak influence of the uncertainty on the process data projected onto the residual subspace. It is worth pointing out that, in this case, the test statistic should be

$$T_{res}^2 = \tilde{y}^T P_{res} \Lambda_{res}^{-1} P_{res}^T \tilde{y},$$

if there exists no numerical problem with Λ_{res}^{-1}.

Pro Nowadays, the amount of process data is huge. People speak about industrial big data. Given a highly dimensional data set Y_N, the PCA technique is helpful to reduce the dimension of the data set so that the data can be well handled in the reduced subspace.

Con This argument sounds reasonable. However, a projection of the process data onto the principal component subspace would, as discussed above, result in poor fault detection performance. Recall that the fault detection performance will be significantly enhanced if the uncertainty in the process data could be considerably reduced. To this end, two strategies will provide us with more efficient solutions:

- When the subspace of the potential faults to be detected is known, a projection of the process data onto this subspace can ensure the required fault detectability on the one hand, and reduce the influence of the uncertainty on the other hand. In fact, the subsequent discussion is a realisation of this solution strategy.
- We restrict fault detection in some defined subspaces. In this case, mapping the process data onto these subspaces can achieve good fault detection performance. It is clear that the improvement of the fault detection performance depends on the mapping algorithm adopted. In the next chapter, we will present approaches for the realisation of this idea towards optimal fault detection in large-scale systems.

Pro On the assumptions that (i) the process under consideration can be modelled in the context of probabilistic PCA (PPCA), which is described by

$$y = Ex + \varepsilon \in \mathcal{R}^m, x \in \mathcal{R}^n, m > n, \tag{13.6}$$

$$\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I), x \sim \mathcal{N}(0, I), rank\,(E) = n, \tag{13.7}$$

(ii) the faults of interest are modelled by

$$y = E(x + f) + \varepsilon \tag{13.8}$$

with fault vector f to be detected, and (iii)

$$\sigma_{\min}(E) >> \sigma_{\varepsilon}, \tag{13.9}$$

a projection onto the principal component subspace given by

$$EE^T + \sigma_{\varepsilon}^2 I = P \begin{bmatrix} \Lambda_{pc} & 0 \\ 0 & \sigma_{\varepsilon}^2 I \end{bmatrix} P^T, P = \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix},$$
$$\Lambda_{pc} = diag\left(\sigma_1^2, \cdots, \sigma_n^2\right), \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2 >> \sigma_{\varepsilon}^2,$$

is reasonable for fault detection and can be applied to build T_{PCA}^2 -test statistic, because

$$P \Lambda_{pc} P^T \approx EE^T.$$

Con At first, it should be emphasised that the PPCA model is, thanks to the separate modelling of correlations among the process variables and measurement noises, well suited for fault detection. On the other hand, it should also be kept in mind that this is achieved at the cost of (considerably) more modelling computations. As an efficient computation tool, the EM algorithm is widely applied for identifying E and σ_{ε}. In our subsequent discussion, we consider the case that the process under consideration could be described using the PPCA model, but the model matrix E is not separately identified. Note that, when matrix E is known, the fault detection problem becomes trivial and can be solved using the standard GLR solution presented in Chap. 3. It is clear that the above PCA algorithm can be successfully applied to fault detection only if the assumptions (13.6)–(13.9) are satisfied. While (13.6) and (13.8) are, more or less, applicable for many technical processes, the assumptions,

$$\varepsilon \sim \mathcal{N}(0, \sigma_{\varepsilon}^2 I), \sigma_{\min}(E) >> \sigma_{\varepsilon},$$

are often unrealistic. This motivates us to discuss about the application of the PCA algorithm under more general conditions. To this end, the above two assumptions are substituted by

$$\varepsilon \sim \mathcal{N}(0, \Sigma_{\varepsilon}), \Sigma_{\varepsilon} > 0, \tag{13.10}$$
$$\lambda_{\min}\left(E^T E\right) + \sigma_{\varepsilon,\min}^2 > \sigma_{\varepsilon,\max}^2, \tag{13.11}$$

where $\lambda_{\min}\left(E^T E\right)$ is the minimum eigenvalue of $E^T E$, and $\sigma_{\varepsilon,\min}, \sigma_{\varepsilon,\max}$ are the minimum and maximum singular value of Σ_{ε} respectively. Recall that the optimal fault detection is achieved, according to the discussion in Sect. 3.4, by "projecting" the measurement y to

$$\bar{y} = E^- y$$

first, and then detecting the fault vector using the standard GLR method. Unfortunately, in the data-driven framework, E is often unknown. Below, we discuss how to apply the PCA algorithm to a successful fault detection using the data matrix $\hat{\Sigma}$. An SVD of $\hat{\Sigma}$ leads to

$$\begin{aligned}\hat{\Sigma} &= P \Lambda P^T, \Lambda = diag\left(\sigma_1^2, \cdots, \sigma_m^2\right), \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_m^2, \\ \Lambda &= \begin{bmatrix} \Lambda_{pc} & 0 \\ 0 & \Lambda_{res} \end{bmatrix}, \Lambda_{pc} = diag\left(\sigma_1^2, \cdots, \sigma_n^2\right), \\ \Lambda_{res} &= diag\left(\sigma_{n+1}^2, \cdots, \sigma_m^2\right) \in \mathcal{R}^{(m-n)\times(m-n)}, \sigma_n^2 >> \sigma_{n+1}^2, \\ P &= \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix} \in \mathcal{R}^{m\times m}, P_{pc} \in \mathcal{R}^{m\times n}.\end{aligned}$$

Note that

$$\hat{\Sigma} \approx EE^T + \Sigma_\varepsilon,$$

and Σ_ε can be further written as

$$\Sigma_\varepsilon = E\Sigma_{\varepsilon,1}E^T + E^\perp \Sigma_{\varepsilon,2}\left(E^\perp\right)^T = \begin{bmatrix} E & E^\perp \end{bmatrix} \begin{bmatrix} \Sigma_{\varepsilon,1} & 0 \\ 0 & \Sigma_{\varepsilon,2} \end{bmatrix} \begin{bmatrix} E^T \\ \left(E^\perp\right)^T \end{bmatrix},$$

where

$$\begin{bmatrix} E^T \\ \left(E^\perp\right)^T \end{bmatrix} \begin{bmatrix} E & E^\perp \end{bmatrix} = \begin{bmatrix} E^T E & 0 \\ 0 & I \end{bmatrix},$$

and $\Sigma_{\varepsilon,1}$, $\Sigma_{\varepsilon,2}$ are some positive definite matrices. As a result, it holds

$$\left(E^\perp\right)^T \hat{\Sigma} E^\perp = \Sigma_{\varepsilon,2}.$$

On the other hand, the assumption (13.11) ensures that

$$\sigma_i^2 = \lambda_i\left(E^T\left(I + \Sigma_{\varepsilon,1}\right)E\right) > \sigma_{i+j}^2 = \sigma_{\varepsilon,j}^2, i = 1, \cdots, n, j = 1, \cdots, m-n,$$

with $\sigma_{\varepsilon,j}, j = 1, \cdots, m-n$, denoting the singular values of $\Sigma_{\varepsilon,2}$. This means,

$$E = P_{pc}T_1, E^\perp = P_{res}T_2 \tag{13.12}$$

and $T_1 \in \mathcal{R}^{n\times n}$, $T_2 \in \mathcal{R}^{(m-n)\times(m-n)}$ are some regular matrices. It yields

$$\begin{aligned} EE^T + \Sigma_\varepsilon &= P_{pc}T_1\left(I + \Sigma_{\varepsilon,1}\right)T_1^T P_{pc}^T + P_{res}T_2\Sigma_{\varepsilon,2}T_2^T P_{res}^T \\ &\Longrightarrow T_1\left(I + \Sigma_{\varepsilon,1}\right)T_1^T = \Lambda_{pc}.\end{aligned}$$

According to the optimal fault detection solution described in Sect. 3.4, the test statistic for an optimal fault detection is given by

$$J = \bar{y}^T \Sigma_{\bar{y}}^{-1} \bar{y}, \bar{y} = E^- \tilde{y}, \Sigma_{\bar{y}} = E^- \left(EE^T + \Sigma_\varepsilon \right) E^{-^T} = I + \Sigma_{\varepsilon,1},$$

which can be further written as

$$\begin{aligned} J &= \bar{y}^T \Sigma_{\bar{y}}^{-1} \bar{y} = \bar{y}^T \left(I + \Sigma_{\varepsilon,1} \right)^{-1} \bar{y} = \left(E^- \tilde{y} \right)^T \left(I + \Sigma_{\varepsilon,1} \right)^{-1} E^- \tilde{y} \\ &= \tilde{y}^T P_{pc} T_1^{-T} \left(I + \Sigma_{\varepsilon,1} \right)^{-1} T_1^{-1} P_{pc}^T \tilde{y} = \tilde{y}^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T \tilde{y}. \end{aligned} \tag{13.13}$$

It is evident that the test statistic J given in (13.13) is exactly the T_{PCA}^2-test statistic used for the PCA fault detection algorithm,

$$T_{PCA}^2 = \tilde{y}^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T \tilde{y}.$$

As a result of the above discussion, the following theorem is proved.

Theorem 13.1 *Given probabilistic model*

$$y = E\,(x + f) + \varepsilon \in \mathcal{R}^m, x \in \mathcal{R}^n, m > n,$$

with the fault vector f to be detected,

$$\varepsilon \sim \mathcal{N}(0, \Sigma_\varepsilon), \Sigma_\varepsilon > 0, x \sim \mathcal{N}(0, I),$$

and suppose that Σ_ε, E are unknown but satisfy

$$\lambda_{\min} \left(E^T E \right) + \sigma_{\varepsilon,\min}^2 > \sigma_{\varepsilon,\max}^2, rank\,(E) = n$$

and the PCA algorithm is applied to the process data $y_i, i = 1, \cdots, N$, where N is (sufficiently) large. Then, the test statistic

$$T_{PCA}^2 = \tilde{y}^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T \tilde{y},$$

and the threshold

$$J_{th,T_{PCA}^2} = \frac{n \left(N^2 - 1 \right)}{N(N - n)} F_\alpha(n, N - n)$$

deliver the best fault detectability for a (given) significance level α.

As a summary of this section and the answers to the two questions formulated at the beginning of this section, we claim that

- the projection technique is a mathematical tool, which can be applied, for instance, for the computation of T^2-test statistic;

- for the fault detection purpose, “projecting” the process data onto the principal component subspace is not necessary, will not bring added-value and should be handled with care;
- a “projection” of the process data onto to the subspace spanned by the fault vectors would improve the fault detection performance. In this context, the PPCA algorithm will result in optimal fault detection, when the process and fault model is of the form and satisfies the conditions, as given in Theorem 13.1.

13.2 Data Centering, Time-Varying Mean and Variance

13.2.1 Problem Description

In most of MVA-based fault detection methods, the first step, both in the offline training and online detection phases, is to center the raw process data. That is, for given process data $y_i, i = 1, \cdots, N$, the mean of the data is first estimated and then subtracted from each measurement, as done in the following computations

$$\bar{y}(N) = \frac{1}{N}\sum_{i=1}^{N} y_i, \bar{y}_i = y_i - \bar{y}(N). \tag{13.14}$$

See, for instance, the PCA algorithm as an example. From the statistical point of view, this step is necessary for building the (estimated) covariance matrix

$$\hat{\Sigma} = \frac{1}{N-1} Y_N Y_N^T, Y_N = \left[\bar{y}_1 \cdots \bar{y}_N \right].$$

Unfortunately, in many publications, this step is not mentioned explicitly and thus less attention has been paid to the questions that may arise:

- Why is centering the process data necessary and under which conditions it could be done?
- What is the consequence of centering the process data for fault detection and diagnosis?
- Which alternative solutions are available and efficient if the conditions for data centering do not hold?

13.2.2 Data Centering: Conditions and Consequence

Given a random vector $y \in \mathcal{R}^m$, its covariance matrix is defined by

$$cov\left(y\right)=\mathcal{E}\left(y-\mathcal{E}y\right)\left(y-\mathcal{E}y\right)^{T}.$$

The sample mean and covariance of y are given by

$$\bar{y}=\frac{1}{N}\sum_{i=1}^{N}y_{i},\ \hat{\Sigma}=\frac{1}{N-1}\sum_{i=1}^{N}\left(y_{i}-\bar{y}\right)\left(y_{i}-\bar{y}\right)^{T}, \tag{13.15}$$

respectively, where $y_i, i=1,\cdots,N$, are the sample data. Now, we consider $y(k)$ as a stochastic process (time series). In general, both

$$\mathcal{E}\left(y(k)\right),cov\left(y\left(k\right)\right)=\mathcal{E}\left(y(k)-\mathcal{E}y(k)\right)\left(y(k)-\mathcal{E}y(k)\right)^{T}$$

are time functions. Only if $y(k)$ is weak-sense stationary (WSS) or (strictly) stationary, $\mathcal{E}\left(y(k)\right)$ and $cov\left(y\left(k\right)\right)$ are time-invariant. On this assumption, $\mathcal{E}\left(y(k)\right)$ and $cov\left(y\left(k\right)\right)$ can be estimated using sample data $y_i, i=1,\cdots,N$, and formula (13.15). It becomes clear that for the computation of the sample covariance matrix $\hat{\Sigma}$, centering the data, as described in (13.14), makes sense only if $y(k)$ is WSS or stationary. This essential condition for the application of data centering is often ignored in applying MVA methods to fault detection. In particular, by those "dynamic versions" of the basic MVA-based fault detection methods like DPCA, DPLS, etc., the term "dynamic" can be misinterpreted as a tool or method to deal with fault detection issues in dynamic processes. In fact, the requirement on WSS or stationarity implies that these methods can only be effectively applied to dynamic systems in the steady state. That is, there exists no change in the mean and variance of the process variables.

On the other hand, a question may arise: why is the computation of the covariance matrix so important? It is well known that the covariance matrix represents variations around the mean value of the process variables. An accurate estimation of the covariance matrix can significantly improve the detectability of those faults that cause changes in the mean of the process variables. For this reason, the T^2-test statistic is, as described in Sect. 3.4, widely used in detecting such faults.

While a correct centering of the process data is often necessary for solving fault detection problems, it should be carefully used in dealing with fault classification issues. For instance, applying k-means method for classifying faults in the mean value of the process variables, data centering may lead to false classification due to the manipulation of the process data and possible loss of useful information about the changes in the mean value.

13.2.3 On Handling Time-Varying Mean

In real applications, the mean of most processes is time-varying. For a successful MVA-based fault detection in such processes, handling of time-varying mean plays an essential role. There are two different strategies to deal with this issue:

- applying data-driven methods for dynamic systems, for instance, the method introduced in Sect. 4.4, to estimating $\mathcal{E}y(k)$ and then to build the residual vector, $y(k) - \hat{y}(k)$, where $\hat{y}(k)$ is an estimate of $\mathcal{E}y(k)$. This scheme is efficient, when a dynamic process driven by certain process input variables is concerned;
- transforming the data into another domain and then handling the detection problem in the new value domain. Typical methods are those time-frequency domain analysis techniques. In the sequel, we focus on this type of methods.

Roughly speaking, given a signal $y(k)$ at different sampling instants, $y\left(k_j\right), j = 0, 1, \cdots, N-1$, it can be transformed into

$$c_i = \sum_{j=0}^{N-1} y\left(k_j\right) \varphi_i(k_j), \tag{13.16}$$

where $\varphi_i(k_j), i = 0, 1, \cdots, N-1$, are basic functions and $c_i, i = 0, 1, \cdots, N-1$, are coefficients. Inversely, $y(k_i)$ can be formally written as

$$y(k_i) = \sum_{j=0}^{N-1} c_j \phi_j(k_i), i = 0, 1, \cdots, N-1, \tag{13.17}$$

where $\phi_j(k_i), j = 0, 1, \cdots, N-1$, define the inverse transform. There are a great number of such discrete transforms for different applications, for instance, the well-known discrete Fourier transform (DFT) or discrete wavelet transform, or more general, the discrete orthonormal transforms, in which $\phi_i(k), \varphi_j(k), k = k_0, \cdots, k_{N-1}$, satisfy

$$\varphi_j(k) = \phi_j^*(k), \sum_{k=0}^{N-1} \phi_i(k)\phi_j^*(k) = \delta_{i-j} = \begin{cases} 1, i = j, \\ 0, i \neq j. \end{cases}$$

It is remarkable that (13.17) can be viewed as an approximation of function $y(k)$ by means of a linear combination of the basic functions $\phi_i(k), i = 0, 1, \cdots, N-1$. Depending on signal properties, the selection of the set of the basic functions can deliver (very) good approximation of $y(k)$.

Let

$$y=\left[y\left(k_0\right)\cdots y\left(k_{N-1}\right)\right], \Phi=\begin{bmatrix}\phi_0(k_0) & \cdots & \phi_0(k_{N-1})\\ \vdots & \vdots & \vdots\\ \phi_{N-1}(k_0) & \cdots & \phi_{N-1}(k_{N-1})\end{bmatrix},$$

$$c=\left[c_0\cdots c_{N-1}\right], \Psi=\begin{bmatrix}\varphi_0(k_0) & \cdots & \varphi_{N-1}(k_0)\\ \vdots & \vdots & \vdots\\ \varphi_0(k_{N-1}) & \cdots & \varphi_{N-1}(k_{N-1})\end{bmatrix}.$$

Equations (13.16)–(13.17) can be written into the following compact form

$$c=y\Psi, y=c\Phi. \tag{13.18}$$

Note that Φ, Ψ are invertible and

$$\Phi\Psi=I.$$

Now, suppose that

$$y(k)=\begin{bmatrix}y_1(k)\\ \vdots\\ y_m(k)\end{bmatrix}\in\mathcal{R}^m$$

is a random vector with

$$y(k)=\mathcal{E}y(k)+\varepsilon\left(k\right),$$

$\varepsilon\left(k\right)\sim\mathcal{N}(0,\Sigma_\varepsilon)$ being white noise. Let $y(k_i), i=1,\cdots,N$, be the recorded process data and denote

$$Y=\left[y(k_1)\cdots y(k_N)\right]=\mathcal{E}\left(Y\right)+\Xi,$$
$$\mathcal{E}\left(Y\right)=\left[\mathcal{E}y(k_1)\cdots\mathcal{E}y(k_N)\right], \Xi=\left[\varepsilon(k_1)\cdots\varepsilon(k_N)\right].$$

By a discrete transform Ψ,

$$\left[\mathcal{E}y_i(k_1)\cdots\mathcal{E}y_i(k_N)\right]\Psi=\left[c_{i,0}\cdots c_{i,N-1}\right], i=1,\cdots,m,$$

we have

$$Y\Psi=\mathcal{E}\left(Y\right)\Psi+\Xi\Psi=C+\Xi\Psi,$$
$$C=\begin{bmatrix}c_{1,0} & \cdots & c_{1,N-1}\\ \vdots & & \vdots\\ c_{m,0} & \cdots & c_{m,N-1}\end{bmatrix}=:\left[C_0\cdots C_{N-1}\right].$$

On the assumption that C is (almost) time-invariant, repeating the above step M times and denoting the resulted C by $C(i), i = 1, \cdots, M$, $Y\Psi$ can be centered to

$$Y_\Psi = Y\Psi - \hat{C} := \Xi_\Psi \approx \Xi\Psi, \hat{C} = \frac{1}{M}\sum_{i=1}^{M} C(i). \tag{13.19}$$

Once the data are centered, fault detection algorithms can be derived. A popular fault detection scheme is to use the data corresponding to a column of $\Xi\Psi$ for fault detection. The idea behind such a scheme is to make use of *a priori* knowledge that the fault will cause evident changes in one of the columns of C, say C_j. A typical example is the fault detection in rotational machines, where a component fault often causes changes at a special frequency of (vibration) sensor signals. Thus, by DFT these changes are transformed into the frequency domain, which are described by the changes in C.

Note that the j-th column of $\Xi\Psi$ is given by

$$\begin{bmatrix} \varepsilon(k_1) & \cdots & \varepsilon(k_N) \end{bmatrix} \begin{bmatrix} \varphi_j(k_1) \\ \vdots \\ \varphi_j(k_N) \end{bmatrix} = \sum_{i=1}^{N} \varepsilon(k_i)\varphi_j(k_i),$$

which is, considering that $\varepsilon(k)$ is a white noise series subject to $\mathcal{N}(0, \Sigma_\varepsilon)$, also normally distributed with zero-mean and constant covariance matrix $\Sigma_{\Psi,j}$, and can be thus written as

$$\bar{\varepsilon}_j := \sum_{i=1}^{N} \varepsilon(k_i)\varphi_j(k_i) \sim \mathcal{N}(0, \Sigma_{\Psi,j}).$$

It yields the following model for detecting changes in C_j:

$$\bar{y}_j := Y \begin{bmatrix} \varphi_j(k_1) \\ \vdots \\ \varphi_j(k_N) \end{bmatrix} = C_j + \bar{\varepsilon}_j, \bar{\varepsilon}_j \sim \mathcal{N}(0, \Sigma_{\Psi,j}).$$

Below, we summarise the above discussions in form of offline training (modelling) and online detection algorithms.

Algorithm 13.1 *Offline training: given (sufficient) process data*

- *Run the discrete transform Ψ for M times and center the transformed data, according to (13.19). The output of this step is:*

$$Y_\Psi(i), i = 1, \cdots, M, \hat{C},$$

where $Y_\Psi(i)$ denotes the result of the i-th computation of Y_Ψ;

- *Form*
$$Y_{\Psi,j} = \left[\, Y_\Psi(1,j) \cdots Y_\Psi(M,j) \,\right], Y_{\Psi,j}Y_{\Psi,j}^* := \hat{\Sigma}_{\Psi,j}$$
with $Y_\Psi(i,j)$ denoting the j-th column of $Y_\Psi(i)$, $i = 1, \cdots, M$;
- *Set the threshold*
$$J_{th,T^2} = \frac{M^2-1}{M(M-1)} F_\alpha(1, M-1).$$

It is evident that $\hat{\Sigma}_{\Psi,j}$ is a sample estimate of $\Sigma_{\Psi,j}$. Therefore, the online detection is realised using the T^2-test statistic as follows.

Algorithm 13.2 *Online detection: given (online) process data $y\left(k_j\right), j = 0, 1, \cdots, N-1$*

- *Run the discrete transform Ψ and center the transformed data, according to (13.19). Let $y_\Psi(j)$ be the j-th column of Y_Ψ;*
- *Compute the T^2-test statistic*
$$J_{T^2} = y_{\Psi,j}^T \hat{\Sigma}_{\Psi,j}^{-1} y_{\Psi,j};$$
- *Check*
$$J_{T^2} - J_{th,T^2}.$$

It should be pointed out that

- the assumption that C is (almost) time-invariant is essential for this detection scheme. This can be achieved by carefully selecting the transforming functions $\varphi_i(k), i = 0, 1, \cdots, N-1$, when knowledge of $\mathcal{E}y(k)$ is known *a priori;*
- often, a fault may cause changes in serval columns of C. In this case, a straightforward extension of the above detection scheme to a fault detection algorithm with multiple test statistics provides us with an effective solution.

13.3 On Detecting Multiplicative Faults and T^2-test Statistic

13.3.1 Problem Description

Generally speaking, multiplicative faults are referred to those undesired changes in system parameters, which may be, for instance, caused by mismatching of operation conditions and control unit parameters or ageing processes in system components. The latter type of the faults are often incipient changes and thus their detection requires highly (fault) sensitive methods. In the MVA-based fault detection framework, given random (measurement) vector $y \in \mathcal{R}^m$, a multiplicative fault is referred to the changes in the covariance matrix. To be specific, suppose

$$y = Hx + \varepsilon, \tag{13.20}$$

where ε represents the measurement noise with zero-mean and covariance matrix Σ_ε, $x \in \mathcal{R}^n$ is the vector of process variables that is uncorrelated with ε. $H \in \mathcal{R}^{m\times n}$ models the process under consideration and takes different values in fault-free or in faulty cases as follows:

$$H = \begin{cases} H_o, & \text{fault-free,} \\ H_f \neq H_o, & \text{faulty.} \end{cases}$$

It is evident that the covariance matrix of the process (measurement) data will change in the faulty case. Although investigations on detecting multiplicative faults by means of different test statistics have been reported, T^2-test statistic is still the mostly used test statistic also for detecting multiplicative faults. In particular, if there is no specification for which type of faults is under consideration, T^2-test statistic is the standard choice. On the other hand, we know, from our discussions in Sect. 3.4, that T^2-test statistic delivers the optimal fault detectability only if additive faults are addressed. The question arises whether the T^2-test statistic would be efficient for detecting multiplicative faults. This will be discussed in the next subsection.

13.3.2 Miss Detection of Multiplicative Faults Using T^2-test Statistic

For the sake of simplicity and without loss of generality, we assume, at first, x is a zero-mean random vector with a unit covariance matrix. It yields

$$cov\,(y) = HH^T + \Sigma_\varepsilon =: \Sigma_y.$$

Let $\hat{\Sigma}_{y,o} > 0$ be a sample estimate of Σ_y in the fault-free operation,

$$\hat{\Sigma}_{y,o} \approx H_o H_o^T + \Sigma_\varepsilon,$$

and build the T^2-test statistic

$$J_{T^2} = y^T \hat{\Sigma}_{y,o}^{-1} y.$$

Let y_f denote the measurement vector in the faulty case. It holds

$$\begin{aligned} cov\left(y_f\right) &= H_f H_f^T + \Sigma_\varepsilon =: \Sigma_f, \\ J_{T^2} &= y_f^T \hat{\Sigma}_{y,o}^{-1} y_f. \end{aligned}$$

For our comparison purpose, we introduce a random vector $\bar{y}$ by transforming y_f to

$$\bar{y} = \Sigma_{y,o}^{1/2} \Sigma_f^{-1/2} y_f.$$

It is evident that $\bar{y}$ and process measurement y in the fault-free operation have the same distribution with the same covariance matrix. Now, re-write J_{T^2} in the faulty operation as

$$\begin{aligned} y_f &= \Sigma_f^{1/2} \Sigma_{y,o}^{-1/2} \bar{y} \Longrightarrow \\ J_{T^2} &= y_f^T \hat{\Sigma}_{y,o}^{-1} y_f = \bar{y}^T \Sigma_{y,o}^{-1/2} \Sigma_f^{1/2} \hat{\Sigma}_{y,o}^{-1} \Sigma_f^{1/2} \Sigma_{y,o}^{-1/2} \bar{y}. \end{aligned}$$

It becomes clear that if

$$\begin{gathered} \Sigma_f^{1/2} \hat{\Sigma}_{y,o}^{-1} \Sigma_f^{1/2} \leq I \Longleftrightarrow \hat{\Sigma}_{y,o}^{-1} \Sigma_f \leq I \\ \Longrightarrow J_{T^2} = y_f^T \hat{\Sigma}_{y,o}^{-1} y_f \leq \bar{y}^T \Sigma_{y,o}^{-1} \bar{y}, \end{gathered}$$

the faulty data will be treated as normal operations and a reliable detection becomes impossible.

Next, we assume that $\mathcal{E}x$ is a constant vector different from zero. It follows from model (13.20) that a multiplicative fault will also cause changes in the mean of the process data. In that case, y_f is given by

$$\begin{aligned} y_f &= H_f x + \varepsilon - H_o \bar{x} = \left(H_f - H_o\right) x + H_o (x - \bar{x}) + \varepsilon \\ &=: \Delta H x + y_o,\, y_o = H_o (x - \bar{x}) + \varepsilon,\, \Delta H = H_f - H_o, \end{aligned}$$

where $\bar{x}$ is the sample mean applied for centering the data, and thus J_{T^2} is subject to

$$\begin{aligned} J_{T^2} &= y_f^T \hat{\Sigma}_{y,o}^{-1} y_f = (\Delta H x + y_o)^T \hat{\Sigma}_{y,o}^{-1} (\Delta H x + y_o) \\ &= y_o^T \hat{\Sigma}_{y,o}^{-1} y_o + x^T \Delta H^T \hat{\Sigma}_{y,o}^{-1} \Delta H x + 2 y_o^T \hat{\Sigma}_{y,o}^{-1} \Delta H x. \end{aligned}$$

That means, the change in J_{T^2} test statistic caused by the multiplicative fault strongly depends on $\Delta H x$. A reliable detection of H_f is realistic if $\|\Delta H x\|$ is considerably large. In other words, an incipient multiplicative fault is hard to be detected using J_{T^2} test statistic.

In summary, it can be concluded that alternative test statistics are needed, in order to achieve a reliable detection of multiplicative faults. This issue will be dealt with in Chap. 15.

13.4 Assessment of Fault Detection Performance

13.4.1 Problem Formulation

In real applications, performance of a fault detection system is generally assessed and quantified by false alarm rate (FAR) and fault detection rate (FDR). In the framework of MVA-based fault detection, FAR, FDR or equivalently missed detection rate

(MDR) are defined in terms of probability, as given in Definitions 2.1–2.3. They are

$$FAR = \Pr\left(J > J_{th} \,|f = 0\right), \tag{13.21}$$

$$MDR = \Pr\left(J \leq J_{th} \,|f \neq 0\right), \tag{13.22}$$

$$FDR = 1 - MDR \tag{13.23}$$

with J, J_{th}, f denoting the test statistic, threshold and faults to be detected, respectively.

Benchmark (case) study is a popular and widely accepted way for the assessment and comparison of different fault detection methods. For instance, Tennessee Eastman Process (TEP) is a mostly used benchmark process for comparison studies on data-driven and MVA-based fault detection methods. Independent of the question whether such benchmark studies are representative or not, the following questions arise:

- are the computation algorithms for FAR and MDR (or FDR) adopted in the benchmark studies correct? Under which conditions can they be applied?
- how far are the computed FAR and MDR (or FDR), considering that all these computation algorithms are based on sample data, confidential?

13.4.2 On FAR and FDR Computation

In benchmark studies, the most popular algorithm for the FAR computation consists of two steps:

- collect data simulated in fault-free operations and compute N samples (values) of J, denoted by $J_1, \cdots, J_N$,
- compute

$$\mathbb{I}\left(J_i\right) = \begin{cases} 1, & \text{if } J_i > J_{th}, \\ 0, & \text{otherwise}, \end{cases}$$

$$FAR = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(J_i\right), \tag{13.24}$$

  and deliver FAR as an estimate for the false alarm rate.

It is clear that FAR is indeed a sample estimate of the probability (13.21), when $J_1, \cdots, J_N$ are independent and identically distributed (i. i. d.). Therefore, under this condition, it is reasonable and fair for the estimation of FAR.

Analogue to this algorithm, the following FDR computation algorithm is also widely adopted:

- collect data simulated during the operation for a *given* fault, say $f = f_o$, and compute N samples (values) of J, denoted by $J_1, \cdots, J_N$,
- compute

$$\mathbb{I}\left(J_i\right) = \begin{cases} 1, & \text{if } J_i > J_{th}, \\ 0, & \text{otherwise,} \end{cases}$$

$$FDR = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(J_i\right), \tag{13.25}$$

and deliver FDR as an estimate for the fault detection rate.

Notice that this algorithm is a sample estimate of FDR only for the case $f = f_o$, which does not, unfortunately, match situations in real applications. Recall that, in general, a fault can be presented in different forms (as time functions), in different directions when the fault is a vector (multiple faults) and in many possible combinations. As a consequence, for computing the probability given in (13.22) (with $f \neq 0$), all these possibilities of the fault should be taken into account. In fact, in the context of fault detection, f can be modelled as a random vector and the probability given in (13.22) should be computed by means of the law of total probability.

In Chaps. 16–17, we will investigate the issues concerning FAR, $FDR\,(MDR)$ as well as mean time to fault detection $(MT2FD)$ in more details. In this framework, the so-called randomised algorithms based computations of FAR, FDR and $MT2FD$ will be proposed, based on which a platform for the performance assessment of fault detection systems will be established.

13.4.3 On Confidential Computations of FAR and FDR

It is clear that the sample estimations of FAR and FDR given in (13.24)–(13.25) are random variables. Thus, the estimation performance depends on various facts, among which the sample number plays an essential role. Below, we briefly study the influence of the sample number on the estimation performance.

In general, our problem can be formulated as follows: given probability

$$p(\gamma) = \Pr\left(J\left(\omega\right) \leq \gamma\right),$$

the confidence level of an estimate $\hat{p}(\gamma)$ for $p(\gamma)$ is the probability, (at least) by which

$$\left|p(\gamma) - \hat{p}(\gamma)\right| < \epsilon,$$

where $\epsilon \in (0, 1)$ is the given accuracy requirement. The confidence level, often denoted by $1 - \delta$ with $\delta \in (0, 1)$, and the required accuracy ϵ are the indicator for the estimation performance. In our study, the sample estimate for $p(\gamma)$ that denotes either FAR or FDR, is given by

$$\hat{p}(\gamma) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(J_i\right), \mathbb{I}\left(J_i\right) = \begin{cases} 1, & \text{if } J_i \in \mathcal{D}_\gamma, \\ 0, & \text{otherwise,} \end{cases} \tag{13.26}$$

where $\mathcal{D}_\gamma$ is the set defined by

$$\mathcal{D}_\gamma = \{J_i \,|J_i > \gamma, i = 1, \cdots N\}, \gamma = J_{th}.$$

The following theorem gives the well-known Hoeffding's inequality.

Theorem 13.2 *Let* $x_i \in [a_i, b_i], i = 1, \cdots, N$, *be i.i.d random variables. For any* $\epsilon > 0$, *it holds*

$$\Pr\left(\sum_{i=1}^{N} x_i - \mathcal{E}\left(\sum_{i=1}^{N} x_i\right) \geq \epsilon\right) \leq e^{-\frac{2\epsilon^2}{\sum\limits_{i=1}^{N}(b_i-a_i)^2}},$$
$$\Pr\left(\sum_{i=1}^{N} x_i - \mathcal{E}\left(\sum_{i=1}^{N} x_i\right) \leq -\epsilon\right) \leq e^{-\frac{2\epsilon^2}{\sum\limits_{i=1}^{N}(b_i-a_i)^2}}.$$

Since

$$\mathcal{E}\hat{p}(\gamma) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{E}\mathbb{I}(J_i) = p(\gamma),$$

it is straightforward that, for $[a_i, b_i] = [0, 1]$, we have the so-called (two-sided) Chernoff bound

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta} \Longrightarrow \Pr\left(\left|p(\gamma) - \hat{p}(\gamma)\right| < \epsilon\right) > 1 - \delta. \tag{13.27}$$

That is, to achieve the required accuracy ϵ with a confidence level $1 - \delta$, N should not be less than $\frac{1}{2\epsilon^2}\log\frac{2}{\delta}$. For a more detailed study, we refer the reader to Chap. 16 as well as to the references given at the end of this chapter.

Example 13.1 *To receive an impression of the needed N according to (13.27), we consider a simple example. For* $\epsilon = 0.01, \delta = 0.001$, *N should not be less than* 16500. *In other words, to achieve an accuracy of* 1% *with a confidence level* 99.9%, *we need at least* 16500 *data. If we have only* 1000 *data available, we can reach, with the confidence level* 99.9% *an estimation accuracy about* 4% *That means, an estimated FAR for* 5% *will statistically make less sense. Unfortunately, such a result can be observed in many (published) benchmark studies.*

13.5 Notes and References

It is the common opinion that the MVA-based fault detection technique is well-established and its applications in the real engineering world are the state of the art. Since years, the research interests in this field have been dedicated to the application of statistical and machine learning techniques to dealing with fault diagnosis issues or applying the existing MVA methods, with slight extensions, to handling industrial big data. In the course of this development, the basic MVA methods often serve as a tool, for instance, for data pre-processing or for building test statistics and determining thresholds. In this context, the three elemental computation steps (algorithms) of standard MVA-based fault detection methods, data projection, data centering and T^2 test statistic, are widely embedded into the different phases of machine learning technique aided fault diagnosis. Due to their importance, misunderstandings or even misuse of these methods may lead to significant degradation in detection performance. This drives us to give a critical review of these algorithms and their applications in detecting faults.

Data centering is a necessary step in most of MVA methods, statistical and machine learning-based fault detection methods and included in the data pre-processing. Unfortunately, in many publications this step has not been explicitly mentioned. In the framework of fault detection, the immediate consequence of this step is that the centered data only contain information about process uncertainties (variations). Although such data and information are essential for fault detection, their use, for example, for fault isolation/classification or root-cause-analysis is questionable. Another aspect to be examined is the pre-conditions for data centering, to which also less attention has been paid. In our review discussion, it has been clearly demonstrated that centering data could be done only if the mean of the data is (nearly) constant. It should be pointed out that the estimation of the mean using (13.1) is efficient only under the condition that the process data are corrupted with (statistical) measurement noise. When deterministic disturbances are present in the process data, computation of mean and then data centering may lose their efficiency for the fault detection purpose and even lead to incorrect detection results.

Time-frequency domain analysis is a well-established technique to deal with fault detection in machines [1]. This technique can also be efficiently applied to solving the problem with time-varying mean in the process measurements. In fact, all those methods can be viewed as data transforms, as generally described in Sect. 13.2 and well-known in signal processing techniques [2, 3]. Generally speaking, such transforms will not improve the signal-to-noise ratio and thus no improvement on fault detectability can be expected. On the other hand, if information about the faults to be detected is available, these techniques can significantly enhance the fault detectability. Indeed, by such a transform the process data are transformed to a feature subspace of the faults, in which the influence of the noises can be remarkably reduced (denoising) and therefore the fault detectability is enhanced.

Projection or more popularly the PCA technique is widely applied in statistical and machine learning techniques to deal with highly dimensional data sets. Our discussions in Sect. 13.1 have demonstrated that

- in the context of fault detection, projection could be a useful technique, for instance, for simplifying problem solutions. But, it cannot, in general, lead to improvement in fault detection performance;
- this is only possible, when information about the faults to be detected is available. In this case, a targeted projection of the process data into the subspace, where the

faults will be present, will reduce the influence of the noises, similar to our above discussion, thanks to the dimension reduction of the measurement subspace;

- if, on the one hand, no information about the faults is available, and a dimensionality reduction is on the other hand necessary, for instance due to large amount of data, the data should be projected into residual subspace, instead of the principal component subspace.

In the next chapter, we will investigate fault detection in large-scale and distributed processes, which is related to the projection technique and can be viewed as an alternative solution to deal with issues of detecting faults in industrial processes with big data.

Detecting multiplicative faults is of considerable practical interest. Unfortunately, as a result of the strong focus on statistical and machine learning based fault detection methods in recent years, less attention has been paid to this topic. It is state of the art that, as a part of the fault detection logic, T^2-test statistic is widely adopted, although it has been, for instance in Sect. 13.3, demonstrated that it is less effective in dealing with multiplicative faults. In Chap. 15, we will address this issue and propose alternative solutions.

Statistical assessment of fault detection performance is not in the focus of MVA-based methods. On the other hand, a fair and reliable assessment of fault detection performance requires statistical and MVA knowledge. In fact, in the context of fault detection, statistical assessment of fault detection performance should be an essential part of any MVA-based methods. Unfortunately, this aspect has received rare attention in research efforts. The standard way of demonstrating the capacity of a proposed fault detection approach and algorithm is to provide some simulation results on a benchmark process or even to use measurement data from a real process. Without doubt, TEP [4] is a useful and representative benchmark process. It can be well used for illustrating the application of a new fault detection algorithm and demonstrating its potential. But, a meaningful assessment of fault detection performance, in particular, in the context of a comparison study, is necessary in a strict and well-defined statistical framework. As summarised in Sect. 13.3, the major deficits in the common simulation and benchmark based performance assessment are:

- insufficient consideration of the faults to be detected in the computation of FDR, and
- insufficient sample number.

For the latter issue, we have introduced Theorem 13.2 given in [5] to illustrate the relation between the sufficient sample number, the estimation accuracy (of FAR and FDR) and the confidence level of the estimation. A more detailed study on this and related issues will be described in Part V.

References

1. R. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications.* John Wiley and Sons, 2011.
2. D. Elliott and K. Rao, *Fast Transforms Algorithms, Analyses, Application.* New York: Academic Press, 1982.
3. S. Wang, "LMS algorithm and discrete orthogonal transforms," *IEEE Trans. on Circuits and Systems*, vol. 38, pp. 949–951, 1991.
4. J. Downs and E. Fogel, "A plant-wide industrial process control problem," *Computers and Chemical Engineering*, vol. 17, pp. 245–255, 1993.
5. R. Tempo, G. Calafiro, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems, Second Edition*. London: Springer, 2013.

Chapter 14
Data-Driven Fault Detection in Large-Scale and Distributed Systems

14.1 Preliminary Knowledge in Network and Graph Theory

Today's large-scale and distributed systems are equipped with communication networks, which connect sub-systems and enable necessary data transmissions among them aiming at optimal system operations. Due to the special role of communication networks in system operations, analysis of influences of network topology on system operation and performance has received considerable research attention in the past decades. In this section, we briefly introduce preliminary knowledge in network and graph theory which are necessary for our subsequent work.

14.1.1 Basic Concepts in Graph Theory

Consider an interconnected process equipped with a communication network. The topology of the network is defined by (i) the nodes in the network and (ii) connections between the nodes. Denote the set of the nodes by $\mathcal{N}$ and the set of connections, which are called edges in the graph theory, by E. Thus, the topology of a network with M nodes can be expressed by

$$\begin{aligned}&\mathcal{G}=(\mathcal{N},E)\,,\mathcal{N}=\{1,\cdots,M\}\,,\\&E=\{(i,j)\,|i,j\in\mathcal{N},\ i\neq j,\ \text{they are networked}\}\,.\end{aligned}$$

We call

$$\mathcal{G}=(\mathcal{N},E)$$

graph. In a graph, edge (i,j) is called directed or undirected, when data transmissions between nodes i and j are performed in one direction or in both directions. A graph is directed or undirected if its edges are directed or undirected, and it is called connected

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_14

if there is a path from any point to any other point in the graph. A simple graph is an undirected graph with neither multiple edges nor loops. Corresponding to real applications, only connected simple graphs are under consideration in our work.

A so-called incidence matrix A associated with a directed graph with M nodes and N edges is defined by

$$A \in \mathcal{R}^{M \times N}, a_{ij} = \begin{cases} 1, \text{ when edge } j \text{ starts from node } i, \\ -1, \text{ when edge } j \text{ ends at node } i, \\ 0, \text{ otherwise.} \end{cases} \tag{14.1}$$

The Laplacian matrix of this graph is defined as

$$L = AA^T. \tag{14.2}$$

We call the minimum length of the paths connecting the node i and node j the distance between the node i and node j and denote it by $d(i, j)$. It is defined

$$d(i, i) = 0.$$

Let d_i be the greatest distance between the node i and any other nodes in the given graph $\mathcal{G} = (\mathcal{N}, E)$. That is

$$d_i = \max_{j \in \mathcal{N}} d(i, j).$$

The diameter d of the graph is defined by

$$d = \max_{i \in \mathcal{N}} d_i. \tag{14.3}$$

In other words, d is the greatest distance between any two nodes in $\mathcal{G}$.

Consider a graph (network) with M nodes. The set of the neighbours of the i-th node consists of all nodes, which are networked with the node i, and is denoted by $\mathcal{N}_i$. That is,

$$\mathcal{N}_i = \{ j \text{ |node } j \text{ is networked with the } i\text{-th node, } j = 1, \cdots, M \}.$$

The number of the edges connected to node i, which is the number of the nodes in $\mathcal{N}_i$ as well, is called degree of node i and denoted by $d_g(i)$.

14.1.2 An Introduction to Distributed Average Consensus

The average consensus method is one of the popular algorithms applied to dealing with distributed optimisation issues in networked systems. In the subsequent sections, we will apply this method for the fault detection purpose in distributed processes equipped with a sensor network. To this end, the basics of average consensus is shortly introduced.

Given a network with M nodes, the average consensus method is an algorithm for the iterative computation of vector $x_i \in \mathcal{R}^{1\times m}$ at the i-th node as follows

$$x_{i,k+1} = w_{ii}x_{i,k} + \sum_{j\in\mathcal{N}_i} w_{ij}x_{j,k}, i = 1, \cdots, M, k = 0, 1, \cdots, \tag{14.4}$$

beginning with some given vector $x_{i,0}$, where $x_{i,k}$ is the computation value of x_i at the k-th iteration, $x_{j,k}$ denotes the computation value of x_j at the k-th iteration and received from the j-th node. Let

$$X_k = \begin{bmatrix} x_{1,k} \\ \vdots \\ x_{M,k} \end{bmatrix} \in \mathcal{R}^{M\times m}, W = \begin{bmatrix} w_{11} & \cdots & w_{1M} \\ \vdots & \ddots & \vdots \\ w_{M1} & \cdots & w_{MM} \end{bmatrix} \in \mathcal{R}^{M\times M},$$

with $w_{ij} = 0$, when $j \notin \mathcal{N}_i, i, j = 1, \cdots, M, i \neq j$. The iteration at all nodes can now be written as

$$X_{k+1} = WX_k \Longrightarrow X_k = W^k X_0, X_0 = \begin{bmatrix} x_{1,0} \\ \vdots \\ x_{M,0} \end{bmatrix}. \tag{14.5}$$

An average consensus is said to be achieved when

$$\lim_{k\to\infty} X_k = \lim_{k\to\infty} W^k X_0 = \frac{\mathbf{1}\mathbf{1}^T}{M}X_0 \Longleftrightarrow \lim_{k\to\infty} W^k = \frac{\mathbf{1}\mathbf{1}^T}{M}. \tag{14.6}$$

In (14.6),

$$\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathcal{R}^M.$$

Note that

$$\frac{\mathbf{1}\mathbf{1}^T}{M}X_0 = \begin{bmatrix} \frac{1}{M}\sum\limits_{i=1}^{M} x_{i,0} \\ \vdots \\ \frac{1}{M}\sum\limits_{i=1}^{M} x_{i,0} \end{bmatrix}, \tag{14.7}$$

which means, the final value at each node is same and equals to the average value of the overall initial values, $x_{i,0}, i = 1, \cdots, M$.

In their highly cited paper on distributed average consensus, Xiao and Boyd have proved the following theorem, which provides us with the necessary and sufficient conditions for the existence of (14.6).

Theorem 14.1 *Given the iteration algorithm (14.5), then (14.6) holds, if and only if*

$$\mathbf{l}^T W = \mathbf{l}^T, \tag{14.8}$$

$$W\mathbf{l} = \mathbf{l}, \tag{14.9}$$

$$\rho\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) < 1, \tag{14.10}$$

where $\rho\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)$ denotes the spectral radius of matrix $W - \frac{\mathbf{l}\mathbf{l}^T}{M}$.

It is well-known that weighting matrices satisfying (14.8)–(14.10) always exist. Since the iteration algorithm (14.5) describes a dynamic system which converges to $\frac{\mathbf{l}\mathbf{l}^T}{M} X_0$, the convergence rate strongly depends on weighting matrix W. Extensive studies on the determination and optimisation of W aiming at the fastest convergence have been reported and some references are also given at the end of this chapter. Below, we briefly describe two methods for constructing W, which are introduced in the papers by Xiao and co-workers.

Let L be the Laplacian matrix of the graph as given in (14.2) and construct

$$W = I - \alpha L. \tag{14.11}$$

It is proved that for some constant α,

$$\rho\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) < 1, \tag{14.12}$$

if and only if

$$0 < \alpha < \frac{2}{\lambda_{\max}(L)}.$$

Moreover,

$$\alpha = \frac{2}{\lambda_{\max}(L) + \lambda_{M-1}(L)} \tag{14.13}$$

gives the minimum value of $\rho\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)$, where $\lambda_{M-1}(L)$ is the $(M-1)$-th largest eigenvalue of L.

Another way of constructing W satisfying (14.8)–(14.10) is

$$W = W^T \in \mathcal{R}^{M \times M}, w_{ij} = \begin{cases} 1 - \frac{d_g(i)}{d_g+1}, i = j, \\ \frac{1}{d_g+1}, j \in \mathcal{N}_i, \\ 0, j \notin \mathcal{N}_i. \end{cases} \tag{14.14}$$

14.2 An Intuitive Average Consensus Based Fault Detection Scheme

14.2.1 System Configuration and Problem Formulation

Suppose that for the purpose of process monitoring, the process under consideration is equipped with a sensor network with M sensor blocks, which are properly networked and modelled by

$$y_i = \mathcal{E}y_i + \varepsilon_i \in \mathcal{R}^m, i = 1, \cdots, M. \tag{14.15}$$

Such a process is schematically sketched in Fig. 14.1. In the context of sensor networks, a sensor block (vector) is called a node. Thus, we consider a sensor network with M nodes.

In the model (14.15), ε_i represents the measurement noise and is assumed to be

$$\varepsilon_i \sim \mathcal{N}(0, \Sigma_i), \Sigma_i > 0, \mathcal{E}\left(\varepsilon_i \varepsilon_j^T\right) = \begin{cases} \Sigma_i, i = j, \\ 0, i \neq j, \end{cases} i = 1, \cdots, M. \tag{14.16}$$

Moreover, it is assumed that

$$\mathcal{E}y_i = \begin{cases} \bar{y}_i, \bar{y}_i = H_i x, \text{ fault-free}, \\ \bar{y}_i + f_i, f_i = H_{i,f} f, \text{ faulty}, \end{cases} i = 1, \cdots, M, \tag{14.17}$$

where $\bar{y}_i = H_i x$ is some unknown constant vector representing the normal process operation measured by the sensor vector (block) y_i located at the i-th node.

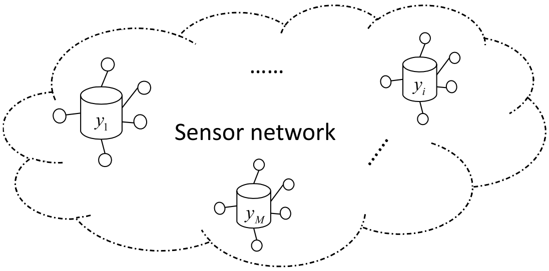


Fig. 14.1 A process equipped with a sensor network consisting of M nodes

Concretely, x is used to denote the (constant) state (vector) of the normal process operation and H_i represents the (unknown) measurement matrix, which can differ at different nodes (location). f_i represents the influence of the (deterministic) process fault f on the i-th sensor block with the unknown distribution matrix $H_{i,f}$.

The M sensor blocks build high degree of measurement redundancy, which can be utilised for a reliable fault detection and reducing the uncertainty due to the measurement noises. To this end, the mean of the measurements at the M sensor blocks is built as follows

$$\bar{y} = \frac{1}{M} \sum_{i=1}^{M} \left(y_i - \bar{y}_i \right).$$

It holds in the fault-free case

$$cov\left(\bar{y}\right) = \mathcal{E}\bar{y}\bar{y}^T = \frac{1}{M^2} \sum_{i=1}^{M} \Sigma_i.$$

It is evident that

$$\begin{aligned} \frac{1}{M}\sigma_{\min} &\leq cov\left(\bar{y}\right) \leq \frac{1}{M}\sigma_{\max}, \\ \sigma_{\min} &= \min\left\{\sigma_{\min}\left(\Sigma_i\right), i = 1, \cdots, M\right\}, \\ \sigma_{\max} &= \max\left\{\sigma_{\max}\left(\Sigma_i\right), i = 1, \cdots, M\right\}. \end{aligned}$$

On the other hand, in the faulty case

$$\mathcal{E}\bar{y} = \frac{1}{M} \sum_{i=1}^{M} \left(y_i - \bar{y}_i \right) = \frac{1}{M} \sum_{i=1}^{M} f_i.$$

On the assumption that

$$\sigma_{\min} > \frac{1}{M}\sigma_{\max}, f_i \approx f_j =: \bar{f}, i, j = 1, \cdots, M,$$

for some fault vector $\bar{f}$, we have

$$cov\left(\bar{y}\right) < \sigma_{\min}, \frac{1}{M} \sum_{i=1}^{M} f_i \approx \bar{f},$$

which shows that increasing the redundancy (the number of the sensor blocks) can reduce the influence of the noise and thus simultaneously improve the fault-to-noise (F2N) ratio defined by

$$F2N = \frac{\left\| \bar{f} \right\|}{\left\| cov\left(\bar{y}\right) \right\|_2}, \left\| \bar{f} \right\|^2 = \bar{f}^T \bar{f}, \left\| cov\left(\bar{y}\right) \right\|_2 = \sigma_{\max}\left(cov\left(\bar{y}\right)\right).$$

The main tasks of detecting the process fault are schematically formulated as

- building the mean of the centred measurement data from the M sensor blocks,

$$\bar{y} = \frac{1}{M} \sum_{i=1}^{M} \left(y_i - \bar{y}_i\right);$$

- building the test statistic

$$J = \bar{y}^T \Sigma^{-1} \bar{y},$$

  where

$$\Sigma = \mathcal{E} \bar{y} \bar{y}^T = \frac{1}{M^2} \sum_{i=1}^{M} \Sigma_i;$$

- setting the threshold

$$J_{th} = \chi_\alpha^2\left(m\right)$$

  for given upper bound of the false alarm rate α, and finally
- defining the (online) detection logic

$$\begin{cases} J - J_{th} > 0 \Longrightarrow \text{faulty}, \\ J - J_{th} \leq 0 \Longrightarrow \text{fault-free}. \end{cases}$$

These tasks should be handled in the distributed and data-driven fashion, in which

- a node has only the access to the local data and communicates with the nodes of its neighbourhood,
- all relevant parameters (matrices) in the model (14.15) are unknown and only process measurement data in the sub-processes located at the sensor nodes are available.

14.2.2 A Basic Average Consensus Algorithm for Fault Detection

We are now in a position to develop a basic scheme of average consensus based fault detection. Like all data-driven methods, the fault detection scheme to be developed consists of a training algorithm and an online implementation algorithm. In addition, the communication protocol should be designed, in order to achieve an average consensus.

On the assumption that sufficient process data have been collected and recorded during the fault-free operation in all sub-systems (nodes), the recorded data are first centered at each node, and the resulted data set at the i-th node is denoted by $Y_i \in \mathcal{R}^{m \times N_i}$, where N_i is the sample number. A key step in the online implementation is a normalisation of the measurement data at each node. To this end,

$$\Sigma_i^{-1/2}, \, \Sigma_i = \frac{1}{N_i - 1} Y_i Y_i^T \in \mathcal{R}^{m \times m}$$

is calculated and saved at each node. Note that such a normalisation has no influence on the fault detection performance and is done locally without communication. The threshold at each node is set to be, for a given FAR upper bound α,

$$J_{th,i} = \chi_\alpha^2 (m), i = 1, \cdots, M.$$

It is clear that the communication protocol should be designed such that matrix W given in (14.5) satisfies conditions (14.8)–(14.10). Below, we assume that these three conditions hold.

The online implementation algorithm, parallel running at the nodes $i, i = 1, \cdots, M$, consists of the following steps.

Algorithm 14.1 *An intuitive average consensus based fault detection:*

Step 0 *Set* $k = 0$, *sample the measurement data* y_i *at each node, and center and normalise it as*

$$y_{i,k}^T = \Sigma_i^{-1/2} (y_i - \bar{y}_i), \, y_{i,k} \in \mathcal{R}^{1 \times m},$$

where $\bar{y}_i$ *is the estimated mean of* y_i, *as defined in the model (14.15)–(14.17) and calculated during the training phase;*

Step 1 *Transmit the data* $y_{i,k}$ *to the neighbours. That is,* $y_{i,k}$ *is sent to the node(s)* $j, i \in \mathcal{N}_j$;

Step 2 *Compute*

$$y_{i,k+1} = w_{ii} y_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij} y_{j,k}, \tag{14.18}$$

and set $k = k + 1$;

Step 3 *Repeat Step 1—Step 2 (iteratively) until*

$$\left\| y_{i,k+1} - y_{i,k} \right\| \leq \gamma,$$

where γ *is a given tolerance constant, and set*

$$\bar{y} = y_{i,k+1}^T;$$

Step 4 *Run*

$$J_i = M \bar{y}^T \bar{y} \tag{14.19}$$

and check

$$J_i - J_{th,i} = M\bar{y}^T\bar{y} - \chi_\alpha^2(m);$$

Step 5 Make decision

$$J_i - J_{th,i} \leq 0 \Longrightarrow fault - free, \; otherwise \; faulty \, and \, alarm;$$

Step 6 (optional) Estimate the (normalised) fault in case of an alarm

$$\hat{f} = \bar{y} \Longrightarrow \mathcal{E}\bar{y} = \frac{1}{M}\sum_{i=1}^{M}\Sigma_i^{-1/2}f_i = \left(\frac{1}{M}\sum_{i=1}^{M}\Sigma_i^{-1/2}H_{i,f}\right)f.$$

We call the above algorithm intuitive average consensus, since it is intuitive to build the average of the measurement data aiming at reducing the influence of the measurement noise. Moreover, the step with the data normalisation is intuitive as well, as done in most of data-driven fault detection schemes. But, this step enables that the training is performed locally and thus without any communications among the nodes. This is a significant advantage of this "intuitive" fault detection algorithm over other distributed fault detection schemes, where distributed learning is needed. On the other hand, it should be noticed that this algorithm does not, in general, deliver an optimal detection solution, as illustrated in our subsequent study.

14.2.3 Performance Analysis and Discussion

Recall that after centering and normalisation the process measurement at the i-th node satisfies (approximately)

$$\Sigma_i^{-1/2}(y_i - \bar{y}_i).$$

Hence,

$$\bar{y} \sim \mathcal{N}\left(0, \frac{1}{M}I_{m\times m}\right),$$

which leads to

$$J_i = M\bar{y}^T\bar{y} \sim \chi^2(m). \tag{14.20}$$

As a result, the threshold is set to be

$$J_{th} = \chi_\alpha^2(m),$$

and the detection algorithm results in the maximal detection rate for the faults modelled in (14.17). It should be emphasised that, although it is realised in the distributed fashion, the fault detection performance in each sub-system (at each node) is identical with the one achievable by running the centralised optimal fault detection scheme.

Note that in the faulty case,

$$\mathcal{E}\bar{y} = \frac{1}{M}\sum_{i=1}^{M} \Sigma_i^{-1/2} f_i = \left(\frac{1}{M}\sum_{i=1}^{M} \Sigma_i^{-1/2} H_{i,f}\right) f,$$

where $\Sigma_i^{-1/2}$ can be interpreted as a weighting matrix that enhances the contribution of f_i to the overall change in the mean caused by the fault, when the variance of the noise at the i-th is weaker.

Remember that the average consensus is an iterative algorithm which delivers the average of the initial values at each node. For our application, this requires, the sampling time should be sufficiently large so that the iteration (14.18) converges and the test statistic satisfies (14.20). Although considerable efforts have been made to accelerate the convergence speed of the iteration towards consensus, the applicability of the distributed fault detection scheme proposed above is strongly limited. Two potential schemes can be followed to deal with this problem: (i) enhancing the fault detectability at cost of delayed fault detection, (ii) increasing the real-time ability for fault detection at cost of detectability of small faults. The following algorithm is proposed for the realisation of the first scheme, whose core is (i) collection of the process measurement data in a time interval distributed (at each node), (ii) building the average of the collected data at each node, and (iii) running the consensus-based fault detection using the average of the collected data over the network. The idea behind it is the reduction of the variance of the measurement data by means of the average building. In details, this algorithm consists of the following steps.

Algorithm 14.2 *A variation of the intuitive average consensus based fault algorithm*

Step 1a *Collect n measurement data* $y_i(1), \cdots, y_i(n)$ *in the time interval* $[t_0, t_1]$ *at each node,*

Step 1b *Parallel to Step 1a, set* $k = 0$, *and center, average and normalise the collected data as*

$$\bar{y}_{i,k}^T = \frac{\Sigma_i^{-1/2}}{n}\sum_{l=1}^{n}\left(y_i(l) - \bar{y}_i\right), \bar{y}_{i,k} \in \mathcal{R}^{1\times m},$$

where $\bar{y}_i$ *is the estimated mean of* y_i, *as defined in the model, (14.15)-(14.17) and calculated during the training phase;*

Step 2 *Transmit the data* $\bar{y}_{i,k}$ *to the neighbours. That is,* $\bar{y}_{i,k}$ *is sent to the node(s)* $j, i \in \mathcal{N}_j$;

Step 3 *Compute*

$$\bar{y}_{i,k+1} = w_{ii}\bar{y}_{i,k} + \sum_{j\in\mathcal{N}_i} w_{ij}\bar{y}_{j,k}, \tag{14.21}$$

and set $k = k + 1$;

Step 4 *Repeat Step 2—Step 3 (iteratively) until*

$$\left\|\bar{y}_{i,k+1} - \bar{y}_{i,k}\right\| \leq \gamma,$$

where γ *is a given tolerance constant, and set*

$$\bar{y} = \bar{y}_{i,k+1}^{T};$$

Step 5 *Run*

$$J_i = nM\bar{y}^T\bar{y} \tag{14.22}$$

and check

$$J_i - J_{th,i} = nM\bar{y}^T\bar{y} - \chi^2_\alpha(m);$$

Step 6 *Make decision*

$$J_i - J_{th,i} \leq 0 \Longrightarrow \text{fault-free, otherwise faulty and alarm;}$$

Step 7 *(optional) In case of an alarm, estimate the (normalised) fault*

$$\hat{f} = \bar{y} \Longrightarrow \mathcal{E}\bar{y} = \frac{1}{M}\sum_{i=1}^{M}\Sigma_i^{-1/2}f_i = \left(\frac{1}{M}\sum_{i=1}^{M}\Sigma_i^{-1/2}H_{i,f}\right)f;$$

Step 8 *In fault-free case, go to Step 1.*

It is clear that the average of the data collected in the time interval, say $[t_0, t_1]$, is processed by the consensus iteration algorithm in the time interval $[t_1, t_2]$. Simultaneously, the measurement data are continuously collected in the time interval $[t_1, t_2]$. At the time instance t_2, fault detection (decision making) is performed. The synchronisation of the data collection and performing the consensus-based fault detection is schematically sketched in Fig. 14.2. In other words, a fault occurring in the time interval $[t_0, t_1]$ can be detected first at the end of the next sampling period $[t_1, t_2]$. On the other hand, the fault detectability is remarkably enhanced, since the covariance matrix of the average of the data collected at each node becomes

$$cov\left(\frac{1}{n}\sum_{l=1}^{n}y_i(l)\right) = \frac{\Sigma_i}{n},$$

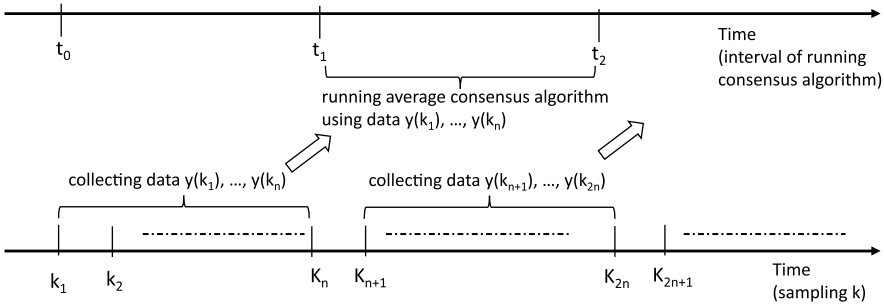

t_0
t_1
t_2
Time
(interval of running
consensus algorithm)
running average consensus algorithm
using data $y(k_1), \ldots, y(k_n)$
collecting data $y(k_1), \ldots, y(k_n)$
collecting data $y(k_{n+1}), \ldots, y(k_{2n})$
k_1
k_2
K_n
K_{n+1}
K_{2n}
K_{2n+1}
Time
(sampling k)

Fig. 14.2 Synchronisation of the data collection and performing consensus algorithm

and the covariance matrix of the average $\bar{y}$ after running the average consensus algorithm is

$$cov\left(\bar{y}\right) = cov\left(\frac{1}{M}\sum_{i=1}^{M}\bar{y}_{i,0}\right) = \frac{1}{nM}I.$$

That means, also small-sized faults can be detected.

In the subsequent section, we will study the realisation of the second scheme, which would lead to increasing the real-time ability for fault detection.

14.3 Practical Algorithms Towards Average Consensus Aided Fault Detection

In this section, we present two algorithms which allow average consensus aided fault detection without the limitation due to the required iteration convergence towards consensus. That means, at each iteration the (online) detection algorithm is performed, instead of waiting for the end of the iteration procedure. It is evident that for a reliable detection by means of T^2 test statistic the covariance matrix of the noise should be well estimated at each iteration. This fact motivates us to update the estimated covariance matrix at each consensus iteration step.

14.3.1 An Algorithm with Distributed Test Statistic Computation

Recall that the test statistic (14.19) in the fault detection algorithm proposed in the last section is computed first after the convergence of the iteration (14.18). The idea of the algorithm proposed in this sub-section is to embed the test statistic computation at each node in the iteration procedure and is thus realised in a distributed manner.

To be specific, we begin with $k = 1$,

$$y_{i,1} = w_{ii} y_{i,0} + \sum_{j \in \mathcal{N}_i} w_{ij} y_{j,0} = W_i Y_0 \Longrightarrow Y_1 = W Y_0, \, Y_1 = \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{M,1} \end{bmatrix},$$
$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_M \end{bmatrix}, \, W_i \in \mathcal{R}^{1 \times M}, \, Y_0 = \begin{bmatrix} y_{1,0} \\ \vdots \\ y_{M,0} \end{bmatrix}, \, y_{i,0} \in \mathcal{R}^{1 \times m}, \, i = 1, \cdots, M.$$

Since for $i = 1, \cdots, M$,

$$y_{i,0}^T = \Sigma_i^{-1/2} \left(y_i - \bar{y}_i \right) \sim \mathcal{N} \left(0, I_{m \times m} \right), \, y_{i,0} \in \mathcal{R}^{1 \times m},$$

and $y_{i,0}^T$ and $y_{j,0}^T$, $i \neq j$, are uncorrelated due to (14.16), it yields

$$y_{i,1}^T \sim \mathcal{N} \left(0, \gamma_{i,1} I_{m \times m} \right),$$
$$\gamma_{i,1} = W_i W_i^T = w_{ii}^2 + \sum_{j \in \mathcal{N}_i} w_{ij}^2, \, i = 1, \cdots, M.$$

For $k = 2$,

$$\begin{aligned} y_{i,2} &= w_{ii} y_{i,1} + \sum_{j \in \mathcal{N}_i} w_{ij} y_{j,1} = W_i W Y_0 = \left(w_{ii} W_i + \sum_{j \in \mathcal{N}_i} w_{ij} W_j \right) Y_0 \\ &= \sum_{l=1}^{M} \left(w_{ii} w_{il} + \sum_{j \in \mathcal{N}_i} w_{ij} w_{jl} \right) y_{l,0} \Longrightarrow Y_2 = W^2 Y_0, \, Y_2 = \begin{bmatrix} y_{1,2} \\ \vdots \\ y_{M,2} \end{bmatrix} \\ &\Longrightarrow y_{i,2}^T \sim \mathcal{N} \left(0, \gamma_{i,2} I_{m \times m} \right), \, i = 1, \cdots, M, \\ \gamma_{i,2} &= \sum_{l=1}^{M} \left(w_{ii} w_{il} + \sum_{j \in \mathcal{N}_i} w_{ij} w_{jl} \right)^2. \end{aligned}$$

The following theorem provides us with a general iteration algorithm.

Theorem 14.2 *Given the process model (14.15)–(14.17) and average consensus iteration (14.18), then*

$$y_{i,k}^T \sim \mathcal{N} \left(0, \gamma_{i,k} I_{m \times m} \right), \tag{14.23}$$

$$V_{i,k} = w_{ii} V_{i,k-1} + \sum_{j \in \mathcal{N}_i} w_{ij} V_{j,k-1}, \, V_{i,0} = e_i, \tag{14.24}$$

$$V_{i,k} = \left[v_{i1,k} \cdots v_{iM,k} \right] \in \mathcal{R}^{1 \times M}, \, k = 0, 1, \cdots,$$

$$\gamma_{i,k} = \sum_{l=1}^{M} \left(w_{ii} v_{il,k-1} + \sum_{j \in \mathcal{N}_i} w_{ij} v_{jl,k-1} \right)^2, i = 1, \cdots, M, \qquad (14.25)$$

where $e_i \in \mathcal{R}^{1\times M}$ *is a vector with all entries equal to* 0 *except for the one at the* i*-th position.*

Proof This theorem can be easily proved by means of mathematical induction. In fact, we have proved that (14.23)–(14.25) are true for $k = 1$. Now, assume that (14.23)–(14.24) hold for $k = q$ and check the result for $k = q + 1$. Since

$$y_{i,q+1} = W_i W^q Y_0 = V_{i,q} Y_0, i = 1, \cdots, M,$$

it is clear

$$V_{i,q} = W_i W^q = W_i W W^{q-1} = w_{ii} V_{i,q-1} + \sum_{j \in \mathcal{N}_i} w_{ij} V_{j,q-1}.$$

Considering

$$\mathcal{E} y_{i,0}^T y_{j,0} = 0, j \neq i, \mathcal{E} y_{i,q+1} = 0,$$

it turns out

$$y_{i,q+1}^T \sim \mathcal{N}\left(0, cov\left(y_{i,q+1}^T\right)\right),$$

$$cov\left(y_{i,q+1}^T\right) = \mathcal{E} Y_0^T V_{i,q}^T V_{i,q} Y_0 = \sum_{l=1}^{M} \left(w_{ii} v_{il,q-1} + \sum_{j \in \mathcal{N}_i} w_{ij} v_{jl,q-1} \right)^2 I.$$

Thus, (14.23)–(14.25) hold for $k = q + 1$.

It follows from this theorem that for the computation of $cov\left(y_{i,k}^T\right)$ at the i-th node, $V_{i,k-1}$ and $V_{j,k-1}, j \in \mathcal{N}_i$, from the neighbours are needed. On the other hand, this computation delivers the covariance matrix of the noise at each iteration and thus allows us to detect fault at each iteration. The corresponding algorithm is summarised as

Algorithm 14.3 *The intuitive average consensus based fault detection algorithm with fault detection during iteration*

Step 0a *Set* $k = 0$, *sample the measurement data* y_i *at the* i*-th node,* $i = 1, \cdots, M$, *and center and normalise it as*

$$y_{i,k}^T = \Sigma_i^{-1/2} \left(y_i - \bar{y}_i\right), y_{i,k} \in \mathcal{R}^{1\times m},$$

where $\bar{y}_i$ *is the estimated mean of* y_i, *as defined in the model (14.15)–(14.17) and calculated during the training phase;*

Step 0b Set

$$V_{i,0} = e_i;$$

Step 1 Transmit the data $y_{i,k}$, $V_{i,k}$ to the neighbours. That is, $y_{i,k}$, $V_{i,k}$ are sent to node(s) $j, i \in \mathcal{N}_j$;

Step 2 Compute

$$y_{i,k+1} = w_{ii} y_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij} y_{j,k}, \tag{14.26}$$

$$V_{i,k+1} = w_{ii} V_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij} V_{j,k}, \tag{14.27}$$

$$\gamma_{i,k+1} = \sum_{l=1}^{M} \left(w_{ii} v_{il,k} + \sum_{j \in \mathcal{N}_i} w_{ij} v_{jl,k} \right)^2; \tag{14.28}$$

Step 3 Run

$$J_i = \frac{1}{\gamma_{i,k+1}} y_{i,k+1} y_{i,k+1}^T \tag{14.29}$$

and check

$$J_i - J_{th,i} = J_i - \chi_\alpha^2 (m);$$

Step 4 Make decision

$$J_i - J_{th,i} \leq 0 \Longrightarrow \textit{fault-free, otherwise faulty;}$$

Step 5 If fault-free, set $k = k + 1$ and repeat Step 1—Step 4 (iteratively) until the next data sampling.

Remark 14.1 *It is evident that $\gamma_{i,k}$ only depends on the weighting matrix W and can be thus offline computed and saved. This can also be done by performing the above average consensus algorithm for the first time. In this way, the needed data transmission is reduced. If the number of the iterations is not large, this is a practical strategy.*

The above fault detection algorithm allows us to detect the fault optimally also during two data samplings. On the other hand, it is of interest to know the convergence rate of $\gamma_{i,k}, i = 1, \cdots, M, k = 0, 1, \cdots$, which is related to the covariance of the noise in the sub-systems and at each iteration. In the literature, much attention has been paid to the convergence rate of the iteration $y_{i,k}$. Although the convergence rate of $y_{i,k}$ and its covariance are strongly relevant, to our best knowledge, few studies have been dedicated to the latter. This motivates our study in the subsequent sub-section. It is remarkable that the result of this work also leads to an online detection algorithm, which requires no computation and communication for the implementation of (14.27)–(14.28).

14.3.2 Analysis of Convergence Rate and an Alternative Fault Detection Algorithm

For our purpose, we first consider $cov\left(y_{i,1}^T\right), i = 1, \cdots, M$. Write $y_{i,1}$ as

$$y_{i,1} = W_i Y_0 = \frac{1}{M}\mathbf{l}^T Y_0 + \Delta_{i,1} Y_0, \Delta_{i,1} = W_i - \frac{1}{M}\mathbf{l}^T.$$

It yields

$$y_{i,1}^T y_{i,1} = Y_0^T \left(\frac{1}{M}\mathbf{l}^T + \Delta_{i,1}\right)^T \left(\frac{1}{M}\mathbf{l}^T + \Delta_{i,1}\right) Y_0.$$

Following the discussion in the previous sub-section, it turns out

$$cov\left(y_{i,1}^T\right) = \sum_{j=1}^{M} \left(\frac{1}{M} + \Delta_{ij,1}\right)^2 I,$$
$$\Delta_{i,1} = \left[\, \Delta_{i1,1} \cdots \Delta_{ij,1} \cdots \Delta_{iM,1} \,\right].$$

Note that

$$\sum_{j=1}^{M} \left(\frac{1}{M} + \Delta_{ij,1}\right)^2 = \left(\frac{1}{M}\mathbf{l}^T + \Delta_{i,1}\right)\left(\frac{1}{M}\mathbf{l}^T + \Delta_{i,1}\right)^T$$
$$= \frac{1}{M} + \Delta_{i,1}\Delta_{i,1}^T,$$

since

$$\Delta_{i,1}\mathbf{l} = \left(W_i - \frac{1}{M}\mathbf{l}^T\right)\mathbf{l} = 0.$$

Therefore, we have

$$cov\left(y_{i,1}^T\right) = \left(\frac{1}{M} + \Delta_{i,1}\Delta_{i,1}^T\right) I. \tag{14.30}$$

Next, we would like to find a upper bound of $cov\left(y_{i,1}^T\right)$ for all $i = 1, \cdots, M$. Noticing that $\Delta_{i,1} = W_i - \frac{1}{M}\mathbf{l}^T$ is a row of matrix $W - \frac{1}{M}\mathbf{l}\mathbf{l}^T$, the value to be found can be determined in terms of $W - \frac{1}{M}\mathbf{l}\mathbf{l}^T$. To this end, we first introduce the following lemma.

Lemma 14.1 *Given a matrix $X \in \mathcal{R}^{n\times m}$ and let $x_i \in \mathcal{R}^{1\times m}$ be the i-th row of $X, i = 1, \cdots, m$, then it holds*

$$x_i x_i^T \leq \sigma_{\max}^2(X) = \|X\|_2^2.$$

Proof Let

$$X = U \Sigma V^T$$

be an SVD of X. It turns out

$$x_i = u_i \Sigma V^T \Longrightarrow x_i x_i^T = u_i \Sigma \Sigma^T u_i^T$$

with u_i as the i-th row of U. Recall that

$$\Sigma \Sigma^T = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ & \ddots & & \vdots \\ \vdots & & \sigma_q^2 & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}, \sigma_{\max}^2 (X) I \geq \Sigma \Sigma^T .$$

It becomes evident that

$$x_i x_i^T \leq \sigma_{\max}^2 (X) .$$

Considering that $\Delta_{i,1}$ is a row of $W - \frac{1}{M}\mathbf{l}\mathbf{l}^T$, according to Lemma 14.1, it holds

$$\forall i \in \{1, \cdots, M\}, \Delta_{i,1}\Delta_{i,1}^T \leq \sigma_{\max}^2 \left(W - \frac{1}{M}\mathbf{l}\mathbf{l}^T \right).$$

Thus,

$$\forall i \in \{1, \cdots, M\}, cov \left(y_{i,1}^T \right) \leq \left(\frac{1}{M} + \sigma_{\max}^2 \left(W - \frac{\mathbf{l}\mathbf{l}^T}{M} \right) \right) I.$$

For a general case with $k \geq 1$, we are able to prove the following theorem.

Theorem 14.3 *Given the process model (14.15)–(14.17) and average consensus iteration (14.18), then*

$$\forall i \in \{1, \cdots, M\}, cov \left(y_{i,k}^T \right) \leq \left(\frac{1}{M} + \left\| W - \frac{\mathbf{l}\mathbf{l}^T}{M} \right\|_2^{2k} \right) I_{m \times m}. \qquad (14.31)$$

Proof Write

$$y_{i,k} = W_i W^{k-1} Y_0, k \geq 1$$

as

$$y_{i,k} = \frac{1}{M}\mathbf{l}^T Y_0 + \Delta_{i,k} Y_0, \Delta_{i,k} = W_i W^{k-1} - \frac{1}{M}\mathbf{l}^T.$$

Note that

$$\begin{gathered}\left(W-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T\right)^2=W^2-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T \Longrightarrow \\ \forall k\geq 1,\left(W-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T\right)^k=W^k-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T, \text{ and} \\ \tfrac{1}{M}W_i\mathbf{l}\mathbf{l}^T=\tfrac{1}{M}\mathbf{l}^T, \tfrac{1}{M}\mathbf{l}^TW^k=\tfrac{1}{M}\mathbf{l}^TW^{k-1}=\tfrac{1}{M}\mathbf{l}^T \Longrightarrow \\ \left(W_i-\tfrac{1}{M}\mathbf{l}^T\right)\left(W^{k-1}-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T\right)=W_iW^{k-1}-\tfrac{1}{M}\mathbf{l}^T.\end{gathered}$$

It turns out

$$\begin{aligned}\varDelta_{i,k}&=W_iW^{k-1}-\frac{1}{M}\mathbf{l}^T=\left(W_i-\frac{1}{M}\mathbf{l}^T\right)\left(W^{k-1}-\frac{1}{M}\mathbf{l}\mathbf{l}^T\right) \\ &=\left(W_i-\frac{1}{M}\mathbf{l}^T\right)\left(W-\frac{1}{M}\mathbf{l}\mathbf{l}^T\right)^{k-1}.\end{aligned}$$

It leads to, along the lines of our discussion on the case $k=1$,

$$\begin{gathered}cov\left(y_{i,k}^T\right)=\left(\tfrac{1}{M}\mathbf{l}^T+\varDelta_{i,k}\right)\left(\tfrac{1}{M}\mathbf{l}^T+\varDelta_{i,k}\right)^T I, \\ \left(\tfrac{1}{M}\mathbf{l}^T+\varDelta_{i,k}\right)\left(\tfrac{1}{M}\mathbf{l}^T+\varDelta_{i,k}\right)^T=\tfrac{1}{M}+\varDelta_{i,k}\varDelta_{i,k}^T+2\tfrac{1}{M}\mathbf{l}^T\varDelta_{i,k}^T, \\ \varDelta_{i,k}\mathbf{l}=\left(W_i-\tfrac{1}{M}\mathbf{l}^T\right)\left(W-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T\right)^{k-1}\mathbf{l}=0 \Longrightarrow \\ \left(\tfrac{1}{M}\mathbf{l}^T+\varDelta_{i,k}\right)\left(\tfrac{1}{M}\mathbf{l}^T+\varDelta_{i,k}\right)^T=\tfrac{1}{M}+\varDelta_{i,k}\varDelta_{i,k}^T,\end{gathered}$$

and further

$$\begin{gathered}\varDelta_{i,k}\varDelta_{i,k}^T=\left(W_i-\tfrac{1}{M}\mathbf{l}^T\right)\left(\left(W-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T\right)\left(W-\tfrac{1}{M}\mathbf{l}\mathbf{l}^T\right)^T\right)^{k-1}\cdot \\ \left(W_i-\tfrac{1}{M}\mathbf{l}^T\right)^T \\ \leq\left(\sigma_{\max}^2\left(W-\tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)\right)^{k-1}\left(W_i-\tfrac{1}{M}\mathbf{l}^T\right)\left(W_i-\tfrac{1}{M}\mathbf{l}^T\right)^T,\end{gathered}$$

which results in

$$cov\left(y_{i,k}^T\right)\leq\frac{1}{M}+\left(\sigma_{\max}^2\left(W-\frac{\mathbf{l}\mathbf{l}^T}{M}\right)\right)^{k-1}\left(W_i-\frac{1}{M}\mathbf{l}^T\right)\left(W_i-\frac{1}{M}\mathbf{l}^T\right)^T.$$

By Lemma 14.1, we finally have

$$\begin{aligned}cov\left(y_{i,k}^T\right)&\leq\left(\frac{1}{M}+\left(\sigma_{\max}^2\left(W-\frac{\mathbf{l}\mathbf{l}^T}{M}\right)\right)^k\right)I_{m\times m} \\ &=\left(\frac{1}{M}+\left\|W-\frac{\mathbf{l}\mathbf{l}^T}{M}\right\|_2^{2k}\right)I_{m\times m}.\end{aligned}$$

According to this theorem,

$$\forall i \in \{1, \cdots, M\}, \lim_{k \to \infty} cov\left(y_{i,k}^T\right) = \frac{1}{M},$$

if

$$\left\| W - \frac{\mathbf{l}\mathbf{l}^T}{M} \right\|_2 = \sigma_{\max}\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) < 1.$$

This is proved in the following theorem on the assumption that W and W^T commute. That is

$$WW^T = W^T W.$$

It is clear that W commutes with W^T if W is symmetric. It is well-known that, if two matrices A, B commute, then

$$\rho(AB) \leq \rho(A)\rho(B).$$

Theorem 14.4 *Given W satisfying the conditions (14.8)–(14.10) given in Theorem 14.1 and commuting with W^T, then*

$$\sigma_{\max}\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) < 1.$$

Proof Since

$$\sigma_{\max}^2\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) = \lambda_{\max}\left(\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)\right)$$

with $\lambda_{\max}$ denoting the maximal eigenvalue of matrix $\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)$, we consider

$$\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) = W^T W + \frac{\mathbf{l}\mathbf{l}^T}{M} - W^T\frac{\mathbf{l}\mathbf{l}^T}{M} - \frac{\mathbf{l}\mathbf{l}^T}{M}W.$$

Recall

$$W^T\mathbf{l} = \mathbf{l},$$

and thus

$$\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) = W^T W - \frac{\mathbf{l}\mathbf{l}^T}{M}.$$

Because

$$W^T W = WW^T$$

leads to

$$\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T \left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right) = \left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T,$$

which means $\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)$ and $\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)^T$ being commute, it holds

$$\begin{aligned}\rho\left(\left(W - \tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\left(W - \tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)\right) &= \lambda_{\max}\left(\left(W - \tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\left(W - \tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)\right)\\ &\le \rho\left(\left(W - \tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)^T\right)\rho\left(\left(W - \tfrac{\mathbf{l}\mathbf{l}^T}{M}\right)\right) < 1.\end{aligned}$$

The theorem is proved.

Theorem 14.3 gives an estimation for the convergence rate of the covariance matrix of $y_{i,k}$, which is of special interest for fault detection using the T^2 test statistic. For our purpose, by the design of the communication protocol, the following optimisation problem is to be solved for W :

$$\begin{gathered}\min_W \sigma_{\max}\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)\\ s.t. W\mathbf{l} = \mathbf{l}, W^T\mathbf{l} = \mathbf{l}.\end{gathered}$$

In their paper on the distributed average consensus, Xiao and Boyd have provided an LMI solution for the above optimisation problem.

Since (14.31) holds for the iterations at all nodes, as a by-product of the above analysis, we propose the following online fault detection algorithm. This algorithm requires no communication of $V_{i,k}$ and computation of (14.27)–(14.28), as needed in the detection algorithm proposed in the last sub-section, but at cost of fault detectability. Below is the algorithm.

Algorithm 14.4 *The intuitive average consensus based fault detection algorithm with fault detection during iteration: the modified version*

Step 0a *Set* $k = 0$, *sample the measurement data* y_i *at the* i*-th node,* $i = 1, \cdots, M$, *and center and normalise it as*

$$y_{i,k}^T = \Sigma_i^{-1/2}\left(y_i - \bar{y}_i\right), y_{i,k} \in \mathcal{R}^{1\times m},$$

where $\bar{y}_i$ *is the estimated mean of* y_i, *as defined in the model (14.15)–(14.17) and calculated during the training phase;*

Step 0b *Compute and save* $\sigma_{\max}^2\left(W - \frac{\mathbf{l}\mathbf{l}^T}{M}\right)$, *and set*

$$\gamma_k = 1;$$

Step 1 Transmit the data $y_{i,k}$ to the neighbours. That is, $y_{i,k}$ are sent to node(s) $j, i \in \mathcal{N}_j$;

Step 2 Compute

$$y_{i,k+1} = w_{ii} y_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij} y_{j,k}, \tag{14.32}$$

$$\gamma_{k+1} = \sigma_{\max}^2 \left(W - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \gamma_k; \tag{14.33}$$

Step 3 Run

$$J_i = \frac{M}{1 + \gamma_{k+1} M} y_{i,k+1} y_{i,k+1}^T \tag{14.34}$$

and check

$$J_i - J_{th,i} = J_i - \chi_\alpha^2 (m);$$

Step 4 Make decision

$$J_i - J_{th,i} \leq 0 \Longrightarrow \textit{fault-free, otherwise faulty;}$$

Step 5 If fault-free, set $k = k + 1$ and repeat Step 1- Step 4 (iteratively) until the next data sampling.

14.3.3 Re-visiting Fault Detection Performance

Remember that the basic idea of the proposed consensus based FD scheme is the utilisation of the (hardware) sensor redundancy to reduce the uncertainty caused by noises in the measurements. This is an intuitive solution to the fault detection problem formulated for distributed processes. On the other hand, we have, in Sect. 3.2, formulated optimal fault detection problem in static processes and given its solution. In this sub-section, we would like to re-visit the FD performance of our intuitive solution in the context of the optimal fault detection formulated in Sect. 3.2.

Consider the process model (14.15)–(14.17). For the simplicity, we assume, in the faulty case,

$$H_{i,f} = I \Longrightarrow \mathcal{E} y_i = \bar{y}_i + f, \ i = 1, \cdots, M. \tag{14.35}$$

Let

$$\tilde{y} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_M \end{bmatrix} = \begin{bmatrix} y_1 - \bar{y}_1 \\ \vdots \\ y_M - \bar{y}_M \end{bmatrix},$$

which yields

$$\begin{aligned} \tilde{y} &= \begin{cases} \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{bmatrix} := \varepsilon \sim \mathcal{N}(0, \varSigma), \text{ fault-free,} \\ \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} f + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{bmatrix} =: H_f f + \varepsilon, \text{ faulty,} \end{cases} \\ \varSigma &= diag\left(\varSigma_1, \cdots, \varSigma_M\right). \end{aligned} \tag{14.36}$$

According to the optimal solution given in Sect. 3.2, given process model (14.36), the optimal test statistic is given by

$$J = \tilde{y}^T \left(H_f^-\right)^T \left(H_f^- \varSigma \left(H_f^-\right)^T\right)^{-1} H_f^- \tilde{y}, H_f^- = \left(H_f^T \varSigma^{-1} H_f\right)^{-1} H_f^T \varSigma^{-1}.$$

Note that

$$\begin{aligned} H_f^- &= \left(\sum_{i=1}^M \varSigma_i^{-1}\right)^{-1} \left[\varSigma_1^{-1} \cdots \varSigma_M^{-1}\right], \\ J &= \left(\sum_{i=1}^M \varSigma_i^{-1} \tilde{y}_i\right)^T \left(\sum_{i=1}^M \varSigma_i^{-1}\right)^{-1} \left(\sum_{i=1}^M \varSigma_i^{-1} \tilde{y}_i\right). \end{aligned} \tag{14.37}$$

Comparing (14.37) with (14.19) makes it clear that the consensus algorithms proposed in this section delivers the optimal fault detection performance only under condition

$$\varSigma_i = I, i = 1, \cdots, M,$$

which can be, obviously, merely satisfied.

In order to understand the proposed consensus-based solution, we slightly modify the process model under consideration to

$$y_i = \begin{cases} H_i\left(x + \varepsilon_i\right), \mathcal{E} y_i = H_i x, \varepsilon_i \sim \mathcal{N}(0, I), \text{ fault-free,} \\ H_i\left(x + \varepsilon_i + f\right), \text{ faulty,} \end{cases} \tag{14.38}$$

$$\begin{aligned} H_i &= \varSigma_i^{1/2}, \mathcal{E} \varepsilon_i^T \varepsilon_j = \delta_{ij}, i, j = 1, \cdots, M, \Longrightarrow \\ \tilde{y} &= \begin{cases} \begin{bmatrix} H_1 \varepsilon_1 \\ \vdots \\ H_M \varepsilon_M \end{bmatrix} =: \varepsilon \sim \mathcal{N}(0, \varSigma), \text{ fault-free,} \\ \begin{bmatrix} H_1\left(\varepsilon_1 + f\right) \\ \vdots \\ H_M\left(\varepsilon_M + f\right) \end{bmatrix} =: H_f f + \varepsilon, \text{ faulty,} \end{cases} \\ \varSigma &= diag\left(\varSigma_1, \cdots, \varSigma_M\right) = diag\left(H_1 H_1^T, \cdots, H_M H_M^T\right). \end{aligned}$$

It is straightforward that

$$\begin{aligned}
H_f^- &= \left(H_f^T \varSigma^{-1} H_f\right)^{-1} H_f^T \varSigma^{-1} = M^{-1} \left[\varSigma_1^{-1/2} \cdots \varSigma_M^{-1/2} \right], \\
J &= \tilde{y}^T \left(H_f^-\right)^T \left(H_f^- \varSigma \left(H_f^-\right)^T\right)^{-1} H_f^- \tilde{y} \\
&= \left(\frac{1}{M} \sum_{i=1}^{M} \varSigma_i^{-1/2} \tilde{y}_i\right)^T M \left(\frac{1}{M} \sum_{i=1}^{M} \varSigma_i^{-1/2} \tilde{y}_i\right),
\end{aligned}$$

which is exactly the test statistic given in (14.19). Thus, the proposed consensus based solution is optimal when the process under consideration can be modelled by (14.38).

At the end of this section, we would like to emphasise that, although our intuitive average consensus algorithms do not result in optimal fault detection performance, they are highly efficient in the context that no data transmissions among the nodes (sub-systems) are required during the training/modelling phase. This also enables online updating the model at each node during system operations without increasing the communication load.

14.4 A General Consensus Based Optimal Fault Detection Scheme

The study in the previous sub-section motivates us to find a consensus based optimal fault detection scheme which can be applied for a general process model form.

14.4.1 Problem Formulation, Solution and Algorithm

We consider again process model (14.15)–(14.17), but

- the dimensions of the measurement vectors in sub-systems can be different. That is
$$y_i \in \mathcal{R}^{m_i}, i = 1, \cdots, M.$$
- And
$$f_i = H_{i,f} f \in \mathcal{R}^{m_i}, H_{i,f} \in \mathcal{R}^{m_i \times k_f}, f \in \mathcal{R}^{k_f}, i = 1, \cdots, M,$$
  with known $H_{i,f}$, which represents the sensor configuration, for instance, $H_{i,f} = H_i$.

Remark 14.2 *In the data-driven framework, the assumption of the known fault model $H_{i,f}$ seems unrealistic, since it is hard to collect data of faulty operations.*

On the other hand, it is helpful in real applications to be able to design a fault detection system which is efficient in detecting a specified fault (pattern) modelled by (designed) $H_{i,f}$. In fact, in order to increase the reliability of fault detection, a bank of such systems can be designed and implemented.

For our purpose, let

$$\tilde{y} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_M \end{bmatrix} = \begin{bmatrix} y_1 - \bar{y}_1 \\ \vdots \\ y_M - \bar{y}_M \end{bmatrix},$$

which leads to

$$\tilde{y} = \begin{cases} \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{bmatrix} =: \varepsilon \sim \mathcal{N}\left(0, \Sigma\right), \text{ fault-free,} \\ \begin{bmatrix} H_{1,f} \\ \vdots \\ H_{M,f} \end{bmatrix} f + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{bmatrix} =: H_f f + \varepsilon, \text{ faulty,} \end{cases} \in \mathcal{R}^{m_\Sigma}, \tag{14.39}$$

$$\Sigma = diag\left(\Sigma_1, \cdots, \Sigma_M\right), m_\Sigma = \sum_{i=1}^{M} m_i.$$

As known from Sect. 3.2 and from our discussion at the end of the previous section, the optimal test statistic is

$$J = \tilde{y}^T \left(H_f^-\right)^T \left(H_f^- \Sigma \left(H_f^-\right)^T\right)^{-1} H_f^- \tilde{y}, H_f^- = \left(H_f^T \Sigma^{-1} H_f\right)^{-1} H_f^T \Sigma^{-1}.$$

Since

$$\left(H_f^- \Sigma \left(H_f^-\right)^T\right)^{-1} = \sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f} \Longrightarrow$$
$$H_f^- = \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f}\right)^{-1} \left[H_{1,f}^T \Sigma_1^{-1} \cdots H_{M,f}^T \Sigma_M^{-1} \right],$$

it holds

$$J = \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} \tilde{y}_i\right)^T \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f}\right)^{-1} \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} \tilde{y}_i\right). \tag{14.40}$$

Moreover, considering

$$H_f^- \varepsilon \sim \mathcal{N}\left(0, H_f^- \Sigma \left(H_f^-\right)^T\right), H_f^- \varepsilon \in \mathcal{R}^{k_f}, \tag{14.41}$$

the corresponding (optimal) threshold is set to be

$$J_{th} = \chi_\alpha^2 \left(k_f\right). \tag{14.42}$$

Also, the MLE of f is given by

$$\hat{f} = H_f^- \tilde{y} = \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f}\right)^{-1} \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} \tilde{y}_i\right). \tag{14.43}$$

It is clear that for the optimal fault detection at each node,

$$\sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f}, \sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} \tilde{y}_i$$

are to be calculated, which can be, for instance, realised using the average consensus algorithm. Below is the data-driven realisation of the average consensus based algorithm for our purpose, which runs in parallel at the nodes $i = 1, \cdots, M$. The algorithm consists of two parts:

- the training phase, in which the computation of

$$\frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f} := \frac{1}{M} \sum_{i=1}^{M} \Psi_i$$

  is realised by an average consensus algorithm,
- the online detection, in which the computation of

$$\frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} \tilde{y}_i := \frac{1}{M} \sum_{i=1}^{M} \varphi_i$$

  is realised by real-time running an average consensus algorithm.

Algorithm 14.5 *Optimal average consensus based fault detection algorithm*

Step 0a *Collect sufficient process data, at each node, for estimating $\Sigma_i, \bar{y}_i$, and compute*

$$\Psi_i = H_{i,f}^T \Sigma_i^{-1} H_{i,f}$$

parallel in all nodes $i = 1, \cdots, M$;

Step 0b *Run an average consensus algorithm, which delivers*

$$\bar{\Psi} = \frac{1}{M}\sum_{i=1}^{M}\Psi_i \tag{14.44}$$

at each node;

Step 1 *Compute*

$$\tilde{y}_i = y_i - \bar{y}_i, \varphi_i = H_{i,f}^T \Sigma_i^{-1}\tilde{y}_i$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 2 *Run an average consensus, which delivers*

$$\bar{\varphi} = \frac{1}{M}\sum_{i=1}^{M}\varphi_i \tag{14.45}$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 3 *Compute*

$$J_i = \bar{\varphi}^T \bar{\Psi}^{-1}\bar{\varphi} \tag{14.46}$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 4 *Check*

$$J_i - J_{th} = J_i - \frac{\chi_\alpha^2\left(k_f\right)}{M}$$

for decision

$$J_i - J_{th} \leq 0 \Longrightarrow \textit{fault-free, otherwise faulty and alarm}$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 5 *In case of an alarm, estimate the fault*

$$\hat{f} = \bar{\Psi}^{-1}\bar{\varphi}$$

in parallel at all nodes $i = 1, \cdots, M$.

14.4.2 Variations of the Algorithm

Analogue to the previous work, we propose two extended variations of the above algorithm aiming at enhancing the fault detection performance.

Enhancing fault detectability using the average of collected data

In Sect. 14.2, we have introduced an algorithm for a consensus-based fault detection using the average of the data collected at each node, in order to enhance the fault detectability. The idea behind this algorithm is to collect the process data during the consensus iteration, as sketched in Fig. 14.2, and to use the average of these data to reduce the variance of the measurement data and so to realise a distributed fault detection with high performance. To be specific, assume that n measurement data $y_i(1), \cdots, y_i(n)$ are collected in the time interval $[t_0, t_1]$ at each node and build the average

$$y_{i,a} = \frac{1}{n} \sum_{l=1}^{n} y_i(l).$$

Note that

$$\mathcal{E} y_{i,a} = \mathcal{E} y_i.$$

As a result, the process model (14.15)–(14.17) is re-written as

$$\begin{aligned} & y_{i,a} = \mathcal{E} y_i + \varepsilon_{i,a} \in \mathcal{R}^{m_i}, i = 1, \cdots, M, \\ & \varepsilon_{i,a} \sim \mathcal{N}\left(0, \Sigma_{i,a}\right), \Sigma_{i,a} = \Sigma_i / n, \mathcal{E}\left(\varepsilon_i^T \varepsilon_j\right) = 0, i \neq j, \\ & \mathcal{E} y_i = \begin{cases} \bar{y}_i, \bar{y}_i = H_i x, \text{ fault-free}, \\ \bar{y}_i + f_i, f_i = H_{i,f} f, \text{ faulty}. \end{cases} \end{aligned}$$

Now, substituting $y_{i,}, \Sigma_i$ in Algorithm 14.5 presented in the previous sub-section by $y_{i,a}, \Sigma_{i,a}$, respectively, results in an extended version of this algorithm with the improved performance. This can be seen clearly by noting that the test statistic given in (14.46) becomes

$$\begin{aligned} J_i &= \left(\frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_{i,a}^{-1} \tilde{y}_i\right)^T \left(\frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_{i,a}^{-1} H_{i,f}\right)^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_{i,a}^{-1} \tilde{y}_i\right) \\ &= n \bar{\varphi}^T \bar{\Psi}^{-1} \bar{\varphi}, \end{aligned}$$

where $\bar{\Psi}, \bar{\varphi}$ are given in (14.44) and (14.45), respectively. Since there is no change in the threshold and the associated FAR, the fault detectability is obviously improved.

Adaptive and reliable fault detection

In real applications, environment conditions around the process under consideration often vary. In many cases, such changes are characterised by time-varying variances of the sensor noises. In case of a slowly time-varying covariance matrix, this problem can be efficiently coped with, for instance, by a recursive algorithm

$$\hat{\Sigma}_{i,k} = (1 - \alpha) \hat{\Sigma}_{i,k-1} + \alpha \tilde{y}_i(k) \tilde{y}_i^T(k), 0 < \alpha < 1,$$

where it is assumed that at the time instant k, sensor data $y_i(k)$ is collected at the i-th node, $i = 1, \cdots, M, k = 0, 1, \cdots, \hat{\Sigma}_{i,k}$ is the update of the estimate for Σ_i, and

$$\tilde{y}_i(k) = y_i(k) - \bar{y}_i.$$

Considering that for our application, $H_{i,f}^T \Sigma_i^{-1}$ and $H_{i,f}^T \Sigma_i^{-1} H_{i,f}$ are needed, the following recursive algorithm for the computation of $\hat{\Sigma}_{i,k}^{-1}$ can be adopted:

$$\begin{aligned}
\hat{\Sigma}_{i,k}^{-1} &= \left((1-\alpha)\, \hat{\Sigma}_{i,k-1} + \alpha \tilde{y}_i(k) \tilde{y}_i^T(k) \right)^{-1} \\
&= \frac{1}{1-\alpha} \left(\hat{\Sigma}_{i,k-1}^{-1} - \frac{\hat{\Sigma}_{i,k-1}^{-1} \tilde{y}_i(k) \tilde{y}_i^T(k) \hat{\Sigma}_{i,k-1}^{-1}}{(1-\alpha) \left(\frac{1}{1-\alpha} \tilde{y}_i^T(k) \hat{\Sigma}_{i,k-1}^{-1} \tilde{y}_i(k) + \frac{1}{\alpha} \right)} \right) \\
&= \frac{1}{1-\alpha} \left(\hat{\Sigma}_{i,k-1}^{-1} - \alpha \frac{\hat{\Sigma}_{i,k-1}^{-1} \tilde{y}_i(k) \tilde{y}_i^T(k) \hat{\Sigma}_{i,k-1}^{-1}}{\left(\alpha \tilde{y}_i^T(k) \hat{\Sigma}_{i,k-1}^{-1} \tilde{y}_i(k) + 1 - \alpha \right)} \right).
\end{aligned} \tag{14.47}$$

Remark 14.3 *There are a great number of algorithms for a recursive update or adaptive estimation of a covariance matrix. Since this computation is not the focus of our study, we just presented the simple algorithm (14.47).*

The implementation of the algorithm to be proposed here follows two phases:

- collecting data and running update of (the inverse of) the covariance matrices at each node according to algorithm (14.47),
- running the consensus-based fault detection algorithm using the (latest) updates of the covariance matrices, which includes a fault detection at each iteration of the consensus algorithm.

The two phases of updating the covariance matrices and running the consensus iteration are performed online and synchronised similar to the one described in Fig. 14.2. Below, we briefly describe the realisation of the consensus based fault detection algorithm.

Assume that at the time instant t_l the estimate $\hat{\Sigma}_i$ is available at the node i, which is delivered by the algorithm (14.47) running during the time interval $\left[t_{l-1}, t_l\right]$ (referred to Fig. 14.2). For the realisation of the test statistic (14.40), a consensus algorithm should be now performed to achieve

$$\sum_{i=1}^{M} H_{i,f}^T \hat{\Sigma}_i^{-1} \tilde{y}_i,$$

and the associated covariance matrix at each node. Note that in the previous fault detection algorithm, the covariance matrix

$$\Psi := \sum_{i=1}^{M} H_{i,f}^T \Sigma_i^{-1} H_{i,f}$$

is computed offline during the training phase. Differently, due to the update in Σ_i^{-1}, Ψ should be now computed online. Denote

$$\phi_i = H_{i,f}^T \hat{\Sigma}_i^{-1} \tilde{y}_i, \Psi_i = H_{i,f}^T \hat{\Sigma}_i^{-1} H_{i,f}.$$

The average consensus algorithm for the computation of

$$\bar{\phi} = \frac{1}{M} \sum_{i=1}^{M} \phi_i, \bar{\Psi} = \frac{1}{M} \sum_{i=1}^{M} \Psi_i$$

is given by

$$\phi_{i,k+1}^T = w_{ii} \phi_{i,k}^T + \sum_{j \in \mathcal{N}_i} w_{ij} \phi_{j,k}^T \Longrightarrow \Phi_{k+1} = W \Phi_k, \tag{14.48}$$

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_M \end{bmatrix}, \Phi_k = \begin{bmatrix} \phi_{1,k}^T \\ \vdots \\ \phi_{M,k}^T \end{bmatrix}, \phi_{i,0} = \phi_i = H_{i,f}^T \hat{\Sigma}_i^{-1} \tilde{y}_i,$$

$$\Psi_{i,k+1} = w_{ii} \Psi_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij} \Psi_{j,k}, i = 1, \cdots, M. \tag{14.49}$$

For our purpose of performing fault detection at each iteration and each node, we check the covariance matrix of $\phi_{i,k+1}^T$. Note that

$$\phi_{i,k+1}^T = W_i W^{k-1} \Phi_0 \sim \mathcal{N}\left(0, \Sigma_{\phi_i,k+1}\right),$$
$$\Sigma_{\phi_i,k+1} = \mathcal{E}\left(\Phi_0^T \left(W^{k-1}\right) W_i^T W_i W^{k-1} \Phi_0\right).$$

Recall that $\tilde{y}_i, \tilde{y}_j, i \neq j$, are uncorrelated. It turns out

$$\mathcal{E}\left(\Phi_0^T \left(W^{k-1}\right) W_i^T W_i W^{k-1} \Phi_0\right) = \sum_{j=1}^{M} v_{i,j,k}^2 H_{j,f}^T \Sigma_j^{-1} H_{j,f}, \tag{14.50}$$
$$W_i W^{k-1} = \left[v_{i,1,k} \cdots v_{i,M,k} \right].$$

It is obvious that (14.50) cannot be exactly achieved by running the consensus algorithm for given W. This motivates us to find an appreciate estimate for the covariance matrix given in (14.50). For our study, we assume that W is doubly stochastic, which means

- all elements of W are nonnegative real numbers and
- conditions (14.8)–(14.9) are satisfied.

It is evident that $\forall k \geq 1, i = 1, \cdots, M,$

$$W_i W^{k-1}\mathbf{1} = W_i\mathbf{1} = 1,$$

and all elements of $W_i W^{k-1}$ are nonnegative real numbers. It yields

$$0 \leq v_{i,j,k} \leq 1 \Longrightarrow v_{i,j,k} \geq v_{i,j,k}^2,\ j = 1, \cdots, M.$$

As a result, we have

$$\mathcal{E}\left(\Phi_0^T \left(W^{k-1}\right) W_i^T W_i W^{k-1} \Phi_0\right) \leq \sum_{j=1}^{M} v_{i,j,k} H_{j,f}^T \Sigma_j^{-1} H_{j,f} = \Psi_{i,k+1}, \qquad (14.51)$$

which is then adopted for detecting faults at the k-th iteration in node i. It should be noticed that as $k \to \infty$,

$$v_{i,j,k} \to \frac{1}{M} \Longrightarrow \sum_{j=1}^{M} v_{i,j,k} H_{j,f}^T \Sigma_j^{-1} H_{j,f} = \frac{1}{M}\sum_{j=1}^{M} H_{j,f}^T \Sigma_j^{-1} H_{j,f}$$
$$> > \sum_{j=1}^{M} v_{i,j,k}^2 H_{j,f}^T \Sigma_j^{-1} H_{j,f} = \frac{1}{M^2}\sum_{j=1}^{M} H_{j,f}^T \Sigma_j^{-1} H_{j,f}.$$

In other words, as the iteration process converges, the estimation for

$$\mathcal{E}\left(\Phi_0^T \left(W^{k-1}\right) W_i^T W_i W^{k-1} \Phi_0\right)$$

by means of (14.51) becomes (very) conservative. A practical solution to deal with this problem is to multiply a correction factor $\alpha(k)$ to $\Psi_{i,k+1}$,

$$\alpha(k)\Psi_{i,k+1} = \alpha(k) \sum_{j=1}^{M} v_{i,j,k} H_{j,f}^T \Sigma_j^{-1} H_{j,f} =: \Psi_{i,k+1,\alpha}, \qquad (14.52)$$

which satisfies

$$\lim_{k\to\infty} \alpha(k) = \frac{1}{M}.$$

Considering that $\rho\left(W - \frac{\mathbf{1}\mathbf{1}^T}{M}\right)$ is the convergence rate of the applied average consensus algorithm,

$$\alpha(k) = \frac{1}{M} + \rho^k\left(W - \frac{\mathbf{1}\mathbf{1}^T}{M}\right) \to \frac{1}{M} \qquad (14.53)$$

could serve for this purpose.

We now summarise the above results in the following algorithm. It is supposed that in the time interval $\left[t_{l-1}, t_l\right]$ the recursive algorithm (14.47) for the computation of $\hat{\Sigma}_i^{-1}$ is performed and delivers $\hat{\Sigma}_i^{-1}$ at the time instant t_l.

Step 0 *Set* $k = 0$ *and*

$$\phi_{i,0}^T = H_{i,f}^T \hat{\Sigma}_i^{-1} \tilde{y}_i, \Psi_{i,0} = H_{i,f}^T \hat{\Sigma}_i^{-1} H_{i,f}$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 1 *Run algorithm (*14.48*)and (*14.49*), and set* $k = k + 1$ *in parallel at all nodes* $i = 1, \cdots, M$;

Step 2 *Compute*

$$\phi_{i,k}^T \Psi_{i,k}^{-1} \phi_{i,k},$$

check and make decision

$$\begin{cases} \phi_{i,k}^T \Psi_{i,k}^{-1} \phi_{i,k} - \chi_\alpha^2 \left(k_f\right) \leq 0, \ fault - free, \\ \phi_{i,k}^T \Psi_{i,k}^{-1} \phi_{i,k} - \chi_\alpha^2 \left(k_f\right) > 0, \ faulty, \end{cases}$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 3 *Repeat Step 1—Step 2 (iteratively) until*

$$\left\| \phi_{i,k+1} - \phi_{i,k} \right\| \leq \gamma,$$

where γ *is a given tolerance constant, set*

$$\bar{\phi} = \phi_{i,k}, \bar{\Psi}^{-1} = \Psi_{i,k}^{-1},$$

and compute

$$J_i = \bar{\phi}^T \bar{\Psi}^{-1} \bar{\phi}$$

in parallel at all nodes $i = 1, \cdots, M$;

Step 4 *Check*

$$J_i - J_{th} = J_i - \frac{\chi_\alpha^2 \left(k_f\right)}{M}$$

for decision

$$J_i - J_{th} \leq 0 \Longrightarrow fault - free, otherwise\, faulty\, and\, alarm$$

in parallel at all nodes $i = 1, \cdots, M$.

Remark 14.4 *If the corrected* $\Psi_{i,k+1}, \Psi_{i,k+1,\alpha}$*, as given in (14.52), is adopted in the algorithm,* $\bar{\Psi}^{-1}$ *in Step 3 will be substituted by*

$$\bar{\Psi}^{-1} = \Psi_{i,k,\alpha}^{-1}$$

and the fault detection logic adopted in Step 4 will be changed to

$$J_i - J_{th} = J_i - \chi_{\alpha}^2 \left(k_f\right).$$

It should be emphasised that the implementation of the above algorithm requires considerable data transmissions, not only the transmission of the measurement data, but also the one of the covariance matrices of the sub-systems, as well as intensive online computation. In comparison, the algorithms proposed in the previous section with normalisation of the measurement in each sub-system by

$$\Sigma_i^{-1/2} \left(y_i - \bar{y}_i\right)$$

save communication and computation efforts, in particular in case that recursive update of the covariance matrices is performed.

14.5 A Distributed Kalman Filter Based Fault Detection Scheme

14.5.1 Problem Formulation

In the previous three sections, we have studied fault detection issues based on the process model (14.15)–(14.17), in which the constant state vector x is assumed. In other words, we have addressed the fault detection issues for static processes. The objective of this section is to extend our average consensus based fault detection scheme to the dynamic processes. To this end, it is supposed a state space model is available with

$$x\left(k+1\right) = Ax(k), x(0) = x_0, x(k) \in \mathcal{R}^n, \tag{14.54}$$

where x is the state vector. Similar to our early study, a sensor network is applied for the purpose of process monitoring. The sensor network under consideration is composed of M nodes and the corresponding graph is connected. It is assumed that the (sensor) measurement vector at the node i is modelled by

$$y_i(kT_{s,i}) = C_i x(kT_{s,i}) + v_i(kT_{s,i}) \in \mathcal{R}^{m_i}, i = 1, \cdots, M. \tag{14.55}$$

In the sensor model (14.55), $v_i(kT_{s,i}) \sim \mathcal{N}\left(0, \Sigma_{v_i}\right)$ represents the measurement noise vector, which is white and uncorrelated with $x(k)$, $v_j(k)$, $j \neq i$, but unknown. In order to deal with practical applications, we suppose that the sampling rate at different sensor nodes could be different. Here, $T_{s,i}$ denotes the sampling time at sensor node i. To simplify our subsequent work, it is assumed that

$$T_{s,i} = \gamma_i T_s, i = 1, \cdots, M,$$

with T_s denoting the sampling time of the process model (14.54) and $\gamma_i \in \{1, 2, \cdots\}$ some integer.

To model the process faults to be detected, the process model (14.54) is extended to

$$x(k+1) = Ax(k) + f(k) \tag{14.56}$$

with $f(k)$ denoting the (deterministic) fault vector. The influence of $f(k)$ on $y(kT_{s,i})$ is modelled by

$$x_f(k+1) = Ax_f(k) + f(k), x_f(0) = 0, \tag{14.57}$$

$$\begin{cases} f(k) = 0, kT_s < t_f, \text{ fault-free,} \\ f(k) \neq 0, kT_s \geq t_f, \text{ faulty,} \end{cases}$$

$$y_i(kT_{s,i}) = C_i x(kT_{s,i}) + f_i(kT_{s,i}) + v_i(kT_{s,i}), \tag{14.58}$$

$$f_i(kT_{s,i}) = C_i x_f\left(kT_{s,i}\right).$$

Here, t_f denotes the time instant, at which the fault f occurs in the process.

In the subsequent work, we would like to propose an average consensus based fault detection scheme using a distributed Kalman filter designed for our sensor network.

14.5.2 Modelling

In order to solve the formulated problem, we first re-build the process and sensor models using the so-called lifting technique, which allows us to deal with multi-sampling rate issues and to apply the average consensus algorithm. To this end, define T as the sampling time of the lifted system model. It should hold

$$T = \eta T_s = \eta_i T_{s,i} = \eta_i \gamma_i T_s, i = 1, \cdots, M, \tag{14.59}$$

with $\eta, \eta_i \in \{1, 2, \cdots\}$ being some integers. Note

$$\eta = \eta_i \gamma_i \Longleftrightarrow \eta / \gamma_i = \eta_i, i = 1, \cdots, M.$$

Remark 14.5 *In the standard lifting technique, η is the smallest common multiplier of $\gamma_i, i = 1, \cdots, M$. For our application, η is selected so that the average consensus* *algorithm converges in the time period T, which can be greater than the smallest common multiplier.*

It is straightforward that the state space model of the lifted system during fault-free operation is given by

$$x\left(\xi+1\right)=A_l x(\xi), x(\xi)=x\left(\xi T\right)=x\left(\xi\eta T_s\right), A_l=A^{\eta}, \tag{14.60}$$

$$y_{i,l}(\xi)=C_{i,l}x(\xi)+v_{i,l}(\xi), y_{i,l}(\xi)=\begin{bmatrix} y_i\left(\xi T\right) \\ y_i\left(\xi T+\gamma_i T_s\right) \\ \vdots \\ y_i\left(\xi T+\left(\eta_i-1\right)\gamma_i T_s\right) \end{bmatrix}, \tag{14.61}$$

$$C_{i,l}=\begin{bmatrix} C_i \\ C_i A^{\gamma_i} \\ \vdots \\ C_i A^{(\eta_i-1)\gamma_i} \end{bmatrix}, v_{i,l}(\xi)=\begin{bmatrix} v_i\left(\xi T\right) \\ v_i\left(\xi T+\gamma_i T_s\right) \\ \vdots \\ v_i\left(\xi T+\left(\eta_i-1\right)\gamma_i T_s\right) \end{bmatrix}, i=1,\cdots,M.$$

In the faulty case, we have

$$x\left(\xi+1\right)=A_l x(\xi)+B_{f,l}\bar{f}_l(\xi), B_{f,l}=\left[\,A^{\eta-1}\ \cdots\ A\ I\,\right],$$

$$y_{i,l}(\xi)=C_{i,l}x(\xi)+\bar{H}_{i,f}\bar{f}_l(\xi)+v_{i,l}(\xi), \bar{f}_l(\xi)=\begin{bmatrix} f(\xi T) \\ f(\xi T+T_s) \\ \vdots \\ f((\xi+1)T-T_s) \end{bmatrix},$$

$$\bar{H}_{i,f}=\begin{bmatrix} 0 & & \cdots & & & & 0 \\ C_i A^{\gamma_i-1} & \cdots & C_i A & C_i & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\ C_i A^{(\eta_i-1)\gamma_i-1} & C_i A^{(\eta_i-1)\gamma_i-2} & \cdots & \cdots & C_i A & C_i & 0 \end{bmatrix},$$

for $i=1,\cdots,M$, which can be summarised as

$$y_l(\xi)=C_l x(\xi)+\bar{H}_f\bar{f}_l(\xi)+v_l(\xi), y_l(\xi)=\begin{bmatrix} y_{1,l}(\xi) \\ \vdots \\ y_{M,l}(\xi) \end{bmatrix}, \tag{14.62}$$

$$C_l=\begin{bmatrix} C_{1,l} \\ \vdots \\ C_{M,l} \end{bmatrix}, \bar{H}_f=\begin{bmatrix} \bar{H}_{1,f} \\ \vdots \\ \bar{H}_{M,f} \end{bmatrix}, v_l(\xi)=\begin{bmatrix} v_{1,l}(\xi) \\ \vdots \\ v_{M,l}(\xi) \end{bmatrix}.$$

Remark 14.6 *To realise a data-driven realisation of distributed Kalman filter based fault detection, the model (14.60)–(14.62) should be identified in the training phase* *using collected process data. This can be performed, for instance, using the data-driven SKR realisation schemes introduced in Chap. 4.*

In the sequel, a distributed Kalman filter based fault detection will be proposed based on the model (14.60)–(14.62).

14.5.3 Kalman Filter Based Optimal Fault Detection: The Basic Idea And the Centralised Solution

For our purpose of fault detection, we first consider a steady (time-invariant) Kalman filter for the residual generation. The delivered residual vector is white and can be used for an optimal fault detection. To be specific, given model (14.60)–(14.62), we have the following Kalman filter based residual generator:

$$\begin{gathered}\hat{x}\,(\xi+1)=A_l\hat{x}(\xi)+L_{kal}r_l(\xi),\\ r_l(\xi)=\begin{bmatrix} r_{1,l}(\xi)\\ \vdots\\ r_{M,l}(\xi)\end{bmatrix}=y_l(\xi)-\hat{y}_l(\xi),\,\hat{y}_l(\xi)=\begin{bmatrix}\hat{y}_{1,l}(\xi)\\ \vdots\\ \hat{y}_{M,l}(\xi)\end{bmatrix}=C_l\hat{x}(\xi),\end{gathered} \tag{14.63}$$

$$\begin{gathered}L_{kal}=A_lPC_l^T\Sigma_r^{-1},\,\Sigma_r=\mathcal{E}\left(r_l(\xi)r_l^T(\xi)\right)=C_lPC_l^T+\Sigma_{v_l},\\ \Sigma_{v_l}=\mathcal{E}\left(v_l(\xi)v_l^T(\xi)\right)=diag\left(\Sigma_{v_{1,l}},\cdots,\Sigma_{v_{M,l}}\right),\,\Sigma_{v_{i,l}}=\begin{bmatrix}\Sigma_{v_i} & & 0\\ & \ddots & \\ 0 & & \Sigma_{v_i}\end{bmatrix}\end{gathered} \tag{14.64}$$

with P as the solution of Riccati equation

$$P=A_lPA_l^T-A_lPC_l^T\left(C_lPC_l^T+\Sigma_{v_l}\right)^{-1}C_lPA_l^T.$$

Thanks to the whiteness property of the residual vector, the fault detection problem can be treated as detecting faults in a statistic process. Since we are interested in an early fault detection, we focus on detecting the fault in its first (lifting) sampling time period T. Remember that $r_l(\xi)$ can be written as

$$r_l(\xi) = \begin{cases} \varepsilon_l \sim \mathcal{N}\left(0, \Sigma_r\right), \text{ fault-free,} \\ \varepsilon_l + \bar{H}_f \bar{f}_l(\xi), \text{ faulty,} \end{cases} \tag{14.65}$$

$$\bar{H}_f = \begin{bmatrix} \bar{H}_{1,f} \\ \vdots \\ \bar{H}_{M,f} \end{bmatrix}, \bar{f}_l(\xi) = \begin{bmatrix} f(\xi T) \\ f(\xi T + T_s) \\ \vdots \\ f((\xi + 1) T - T_s) \end{bmatrix},$$

$$\bar{H}_{i,f} = \begin{bmatrix} 0 & & \cdots & & & & 0 \\ C_i A^{\gamma_i - 1} & \cdots & C_i A & C_i & 0 & & \cdots 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\ C_i A^{(\eta_i - 1)\gamma_i - 1} & C_i A^{(\eta_i - 1)\gamma_i - 2} & \cdots & \cdots & C_i A & C_i & 0 \end{bmatrix}.$$

It is evident that the first row and column blocks of $\bar{H}_{i,f}$ equal to zero. That means, the residual vector at the time instance ξT is not affected by the fault vector, and $r_l(\xi)$ is independent of $f((\xi + 1) T - T_s)$. In order to be able to detect the fault in a full sampling time period T, we re-define the (lifted) fault vector as

$$f_l(\xi) := \begin{bmatrix} f(\xi T - T_s) \\ f(\xi T) \\ \vdots \\ f((\xi + 1) T - 2T_s) \end{bmatrix}.$$

Recall

$$x(\xi T) = Ax(\xi T - T_s) + f(\xi T - T_s).$$

Let

$$x_0(\xi) := Ax(\xi T - T_s) \Longrightarrow x(\xi T) = x_0(\xi) + f(\xi T - T_s),$$

that is, $x_0(\xi)$ is independent of $f(\xi T - T_s)$. It holds

$$y_l(\xi) = C_l x_0(\xi) + H_f f_l(\xi) + v_l(\xi), H_f = \begin{bmatrix} H_{1,f} \\ \vdots \\ H_{M,f} \end{bmatrix},$$

$$H_{i,f} = \begin{bmatrix} C_i & & \cdots & & & \\ C_i A^{\gamma_i} & \cdots & C_i A & C_i & 0 & \cdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots \\ C_i A^{(\eta_i - 1)\gamma_i} & C_i A^{(\eta_i - 1)\gamma_i - 1} & \cdots & \cdots & C_i A & C_i \end{bmatrix}.$$

Notice that for $f(\xi T - T_s) = 0$,

$$x(\xi T) = x_0(\xi),$$

and $\hat{x}(\xi)$ is independent of $f(\xi T - T_s)$.

Now, we assume, without loss of generality, that the fault occurs, for the first time, in the time interval $[\xi T - T_s, (\xi + 1) T - T_s)$. It follows from the above discussion that the detection model (14.65) can be equivalently written as

$$r_l(\xi) = \begin{cases} \varepsilon_l \sim \mathcal{N}(0, \Sigma_r), \text{ fault-free,} \\ \varepsilon_l + H_f f_l(\xi), \text{ faulty,} \end{cases} \in \mathcal{R}^{\varsigma}, \varsigma = \sum_{i=1}^{M} \eta_i m_i. \tag{14.66}$$

It should be emphasised that the model (14.66) describes the residual dynamics for the case

$$f_l(\xi - 1) = 0, f_l(\xi) \neq 0.$$

Next, the test statistic and threshold are to be determined. It is clear that

$$J = r_l^T(\xi) \left(H_f^T \Sigma_r^{-1}\right)^T \left(H_f^T \Sigma_r^{-1} H_f\right)^{-1} H_f^T \Sigma_r^{-1} r_l(\xi), \tag{14.67}$$

$$J_{th} = \chi_\alpha^2 \left(k_f\right), k_f = \eta n, \tag{14.68}$$

perform an optimal fault detection.

Remark 14.7 *It should be noticed that, although ηn is generally much more smaller than ς for $M \geq n$, it could be a large number, which may result in heavy computation load. On the other hand, detecting incipient faults is a challenging issue and of considerable practical interest. Typically, incipient faults are small and change slowly. In this context, it is reasonable to assume $f_l(\xi)$ is almost a constant vector in the time interval $[\xi T - T_s, (\xi + 1) T - T_s)$,*

$$f(t) \approx f, t \in [\xi T - T_s, (\xi + 1) T - T_s).$$

On this assumption, we have

$$H_{i,f} f_l(\xi) = \begin{bmatrix} C_i \\ \sum_{j=0}^{\gamma_i} C_i A^j \\ \vdots \\ \sum_{j=0}^{(\eta_i - 1)\gamma_i} C_i A^j \end{bmatrix} f, k_f = n.$$

As demonstrated in the following example, in this case the threshold can be set considerably lower than the one given in (14.68) for the same FAR.

Example 14.1 *Given the residual model (14.66) and assume that*

$$f(k) = f \in \mathcal{R}^n$$

is a constant vector. It leads to

$$H_f f_l(\xi) = \begin{bmatrix} H_1 \\ \vdots \\ H_M \end{bmatrix} f =: Hf,\, H_i = \begin{bmatrix} C_i \\ \sum\limits_{j=0}^{\gamma_i} C_i A^j \\ \vdots \\ \sum\limits_{j=0}^{(\eta_i - 1)\gamma_i} C_i A^j \end{bmatrix}, i = 1, \cdots, M.$$

Now, applying the GLR method results in the following test statistic and threshold setting

$$\begin{aligned} J &= r_l^T(\xi) \left(H^T \Sigma_r^{-1}\right)^T \left(H^T \Sigma_r^{-1} H\right)^{-1} H^T \Sigma_r^{-1} r_l(\xi), \\ J_{th} &= \chi_\alpha^2(n). \end{aligned}$$

It is clear that

$$\chi_\alpha^2(n) << \chi_\alpha^2(\eta n).$$

Thus, the threshold in our example is much lower than the one given in (14.68) for the same FAR α.

When a fault estimation is additionally required, the MLE of $f_l(\xi)$ is given by

$$\hat{f}_l(\xi) = H_f^- r_l(\xi) = \left(H_f^T \Sigma_r^{-1} H_f\right)^{-1} H_f^T \Sigma_r^{-1} r_l(\xi). \tag{14.69}$$

14.5.4 A Distributed Kalman Filter-Based Optimal Fault Detection Scheme

The distributed realisation of Kalman filter based residual generator (14.63)–(14.64) and the test statistic (14.67) as well as fault estimation (14.69) consists of two phases: (i) distributed offline training (learning), and (ii) distributed online fault detection.

Distributed offline training

On the assumption that at the i-th node, $i \in \{1, \cdots, M\}$,

- the process model (14.60) with the sampling time T as well as η,
- C_i and sufficient local measurement data $y_i(k)$

are available, Σ_{v_i}, $H_{i,f}$ are first computed at the i-th node, before starting with a consensus based training. The objective of the offline training is to achieve a consensus at all nodes with the needed parameters (matrices) so that the identical Kalman filter and test statistic can be performed at all these nodes. To this end, each node should have, according to (14.63)–(14.64), (14.67) and (14.69), the following parameters

$$C_l^T \Sigma_r^{-1}, C_l^T \Sigma_r^{-1} C_l, H_f^T \Sigma_r^{-1}, H_f^T \Sigma_r^{-1} H_f.$$

It follows from the identity

$$\Sigma_r^{-1} = \left(C_l P C_l^T + \Sigma_{v_l}\right)^{-1} = \Sigma_{v_l}^{-1} - \Sigma_{v_l}^{-1} C_l \left(P^{-1} + C_l^T \Sigma_{v_l}^{-1} C_l\right)^{-1} C_l^T \Sigma_{v_l}^{-1}$$

that

$$\begin{aligned}
C_l^T \Sigma_r^{-1} &= P^{-1}\left(P^{-1} + C_l^T \Sigma_{v_l}^{-1} C_l\right)^{-1} C_l^T \Sigma_{v_l}^{-1}, \\
C_l^T \Sigma_{v_l}^{-1} &= \left[\, C_{1,l}^T \Sigma_{v_{1,l}}^{-1} \cdots C_{M,l}^T \Sigma_{v_{M,l}}^{-1} \,\right], C_l^T \Sigma_{v_l}^{-1} C_l = \sum_{i=1}^{M} C_{i,l}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}, \\
C_l^T \Sigma_r^{-1} C_l &= P^{-1}\left(P^{-1} + C_l^T \Sigma_{v_l}^{-1} C_l\right)^{-1} C_l^T \Sigma_{v_l}^{-1} C_l, \\
H_f^T \Sigma_r^{-1} &= H_f^T \Sigma_{v_l}^{-1} - H_f^T \Sigma_{v_l}^{-1} C_l \left(P^{-1} + C_l^T \Sigma_{v_l}^{-1} C_l\right)^{-1} C_l^T \Sigma_{v_l}^{-1}, \\
H_f^T \Sigma_{v_l}^{-1} &= \left[\, H_{1,f}^T \Sigma_{v_{1,l}}^{-1} \cdots H_{M,f}^T \Sigma_{v_{M,l}}^{-1} \,\right], H_f^T \Sigma_{v_l}^{-1} C_l = \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}, \\
H_f^T \Sigma_r^{-1} H_f &= H_f^T \Sigma_{v_l}^{-1} H_f - H_f^T \Sigma_{v_l}^{-1} C_l \left(P^{-1} + C_l^T \Sigma_{v_l}^{-1} C_l\right)^{-1} C_l^T \Sigma_{v_l}^{-1} H_f, \\
H_f^T \Sigma_{v_l}^{-1} H_f &= \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} H_{i,f}.
\end{aligned}$$

It becomes obvious that

$$\begin{aligned}
C_l^T \Sigma_{v_l}^{-1} C_l &= \sum_{i=1}^{M} C_{i,l}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}, \; H_f^T \Sigma_{v_l}^{-1} H_f = \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} H_{i,f}, \\
H_f^T \Sigma_{v_l}^{-1} C_l &= \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}
\end{aligned}$$

are to be found by means of the consensus algorithm, in order to compute

$$\begin{aligned}
P &= A_l P A_l^T - A_l P C_l^T \left(C_l P C_l^T + \Sigma_{v_l}\right)^{-1} C_l P A_l \Longleftrightarrow \\
P &= A_l \left(P^{-1} + \sum_{i=1}^{M} C_{i,l}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}\right)^{-1} A_l^T,
\end{aligned}$$

$$\left(H_f^T \Sigma_r^{-1} H_f\right)^{-1} = \begin{pmatrix} \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} H_{i,f} - \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} C_{i,l} \cdot \\ \left(P^{-1} + \sum_{i=1}^{M} C_{i,l}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}\right)^{-1} \left(\sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}\right)^T \end{pmatrix}^{-1}$$

at each node. The following algorithm is proposed for our purpose.

Algorithm 14.6 *The training algorithm for distributed Kalman filter based fault detection*

Step 0: *Set* $k = 0$ *and compute*

$$\Sigma_{C_i,k} = C_{i,l}^T \Sigma_{v_{i,l}}^{-1} C_{i,l},\ \Sigma_{H_i,k} = H_{i,f}^T \Sigma_{v_{i,l}}^{-1} H_{i,f},\ \Sigma_{HC_i,k} = H_{i,f}^T \Sigma_{v_{i,l}}^{-1} C_{i,l}$$

in parallel at nodes $i = 1, \cdots, M$;

Step 1: *Start an average consensus algorithm to compute*

$$\bar{\Sigma}_{C,k} = \frac{1}{M} \sum_{i=1}^{M} C_{i,l}^T \Sigma_{v_{i,l}}^{-1} C_{i,l},\ \bar{\Sigma}_{H,k} = \frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} H_{i,f},$$

$$\bar{\Sigma}_{HC_i,k} = \frac{1}{M} \sum_{i=1}^{M} H_{i,f}^T \Sigma_{v_{i,l}}^{-1} C_{i,l};$$

Step 2: *Solve*

$$P = A_l \left(P^{-1} + M \bar{\Sigma}_{C,k} \right)^{-1} A_l^T$$

for P *in parallel at nodes* $i = 1, \cdots, M$;
Step 3: *Calculate*

$$\bar{\Psi} = \bar{\Sigma}_{H,k} - M \bar{\Sigma}_{HC_i,k} \left(P^{-1} + M \bar{\Sigma}_{C,k} \right)^{-1} \bar{\Sigma}_{HC_i,k}^T$$

in parallel at nodes $i = 1, \cdots, M$;
Step 4: *Output*

$$\left(P^{-1} + C_l^T \Sigma_{v_l}^{-1} C_l \right)^{-1} = \left(P^{-1} + M \bar{\Sigma}_{C,k} \right)^{-1}, \bar{\Psi} = \frac{1}{M} H_f^T \Sigma_r^{-1} H_f,$$

$$\bar{\Sigma}_{HC_i,k} = \frac{1}{M} H_f^T \Sigma_{v_l}^{-1} C_l$$

at nodes $i = 1, \cdots, M$.

Distributed online implementation
Re-write the distributed realisation of Kalman filter based residual generator (14.63)–(14.64), the test statistic (14.67) and the fault estimate (14.69) into

$$\begin{aligned}
\hat{x}_i\left(\xi+1\right) &= A_l\hat{x}_i(\xi) + L_{kal}r(\xi) = A_l\hat{x}_i(\xi) + A_lPC_l^T\Sigma_r^{-1}r(\xi)\\
&= A_l\hat{x}_i(\xi) + A_l\left(P^{-1}+C_l^T\Sigma_{v_l}^{-1}C_l\right)^{-1}\sum_{j=1}^{M}C_{j,l}^T\Sigma_{v_{j,l}}^{-1}r_j(\xi),\\
r_j(\xi) &= y_{j,l}(\xi) - \hat{y}_{j,l}(\xi),\, \hat{y}_{j,l}(\xi) = C_{j,l}\hat{x}_j(\xi),\\
J_i &= r^T(\xi)\left(H_f^T\Sigma_r^{-1}\right)^T\left(H_f^T\Sigma_r^{-1}H_f\right)^{-1}H_f^T\Sigma_r^{-1}r(\xi)\\
&= r_J^T(\xi)\left(H_f^T\Sigma_r^{-1}H_f\right)^{-1}r_J(\xi),\\
r_J(\xi) &= H_f^T\Sigma_r^{-1}r(\xi)\\
&= \sum_{j=1}^{M}\left(H_{j,f}^T - H_f^T\Sigma_{v_l}^{-1}C_l\left(P^{-1}+C_l^T\Sigma_{v_l}^{-1}C_l\right)^{-1}C_{j,l}^T\right)\Sigma_{v_{j,l}}^{-1}r_j(\xi),\\
\hat{f}_l(\xi) &= \left(H_f^T\Sigma_r^{-1}H_f\right)^{-1}r_J(\xi).
\end{aligned}$$

Here, $\hat{x}_i(\xi)$ denotes the estimate of the state vector $x(\xi)$ delivered by the Kalman filter running at the node i. Since identical Kalman filters are realised at all nodes, it holds

$$\hat{x}_i(\xi) = \hat{x}_j(\xi), i, j = 1, \cdots, M.$$

As a result, we can run the following algorithm for an online fault detection in the time interval $[\xi T, (\xi+1)T)$.

Algorithm 14.7 *The online implementation algorithm for distributed Kalman filter based fault detection*

Step 0: *Compute*

$$\begin{aligned}
r_{j,KF}(\xi) &= C_{j,l}^T\Sigma_{v_{j,l}}^{-1}r_j(\xi),\\
r_{j,J}(\xi) &= \left(H_{j,f}^T - H_f^T\Sigma_{v_l}^{-1}C_l\left(P^{-1}+C_l^T\Sigma_{v_l}^{-1}C_l\right)^{-1}C_{j,l}^T\right)\Sigma_{v_{j,l}}^{-1}r_j(\xi)
\end{aligned}$$

in parallel at nodes $j = 1, \cdots, M$;

Step 1: *Start an average consensus algorithm to compute*

$$\bar{r}_{KF}(\xi) = \frac{1}{M}\sum_{j=1}^{M}r_{j,KF}(\xi), \bar{r}_J(\xi) = \frac{1}{M}\sum_{i=1}^{M}r_{j,J}(\xi);$$

Step 2: *Calculate*

$$\begin{aligned}
\hat{x}_j\left(\xi+1\right) &= A_l\hat{x}_j(\xi) + A_l\left(P^{-1}+C_l^T\Sigma_{v_l}^{-1}C_l\right)^{-1}M\bar{r}_{KF}(\xi),\\
J_j &= M\bar{r}_J^T(\xi)\bar{\Psi}^{-1}\bar{r}_J(\xi)
\end{aligned}$$

in parallel at nodes $j = 1, \cdots, M$;

Step 3a: *Check*

$$J_j - J_{th} = J_j - \chi^2_\alpha \left(k_f\right)$$

for decision

$$J_j - J_{th} \leq 0 \Longrightarrow fault - free,\ otherwise\, faulty\, and\, alarm$$

at all nodes $j = 1, \cdots, M$. *In the faulty case,*

Step 3b (optional) *Calculate*

$$\hat{f}_l(\xi) = \bar{\Psi}^{-1}\bar{r}_J(\xi);$$

Step 4: *Output*

$$\hat{x}_j\left(\xi + 1\right), \textit{and in faulty case alarm and, optionally, } \hat{f}_l(\xi)$$

at nodes $j = 1, \cdots, M$.

The synchronisation of online collecting data and performing the above consensus based fault detection algorithm is analogue to the workflow sketched in Fig. 14.2.

14.6 Correlation-Based Fault Detection Schemes and Algorithms

14.6.1 Problem Formulation

The process model considered in this section is a large-scale process consisting of M sub-systems, which are connected by a communication network, as sketched in Fig. 14.3.

It is supposed that each sub-system is a node of the network and modelled by

$$y_i = \mathcal{E}y_i + \varepsilon_i \in \mathcal{R}^{m_i}, i = 1, \cdots, M. \tag{14.70}$$

Here, ε_i represents the (local) measurement noise satisfying

$$\varepsilon_i \sim \mathcal{N}\left(0, \Sigma_{ii}\right), \Sigma_{ii} > 0, \tag{14.71}$$

and

$$\mathcal{E}y_i = \begin{cases} \bar{y}_i, & \text{fault-free,} \\ \bar{y}_i + f_i, & \text{faulty,} \end{cases} \tag{14.72}$$

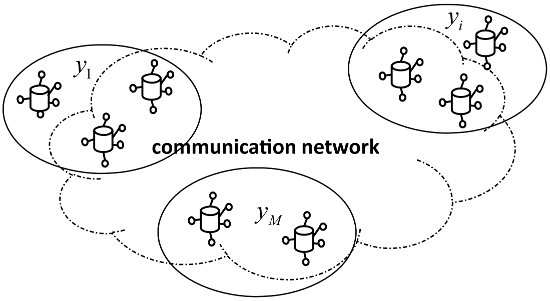


Fig. 14.3 A process with M sub-systems and a communication network

where $\bar{y}_i$ is some unknown constant vector representing the normal process operation and f_i represents the fault to be detected in the i-th sub-system. The overall process model for the normal process operation is given by

$$y = \mathcal{E}y + \varepsilon \in \mathcal{R}^m, m = \sum_{i=1}^{M} m_i, y = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}, \mathcal{E}y = \begin{bmatrix} \mathcal{E}y_1 \\ \vdots \\ \mathcal{E}y_M \end{bmatrix},$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{bmatrix} \sim \mathcal{N}(0, \Sigma), \Sigma = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1i} & \cdots & \Sigma_{1M} \\ \vdots & \ddots & & & \vdots \\ \Sigma_{i1} & \cdots & \Sigma_{ii} & \cdots & \Sigma_{iM} \\ \vdots & & \vdots & \ddots & \vdots \\ \Sigma_{M1} & \cdots & \Sigma_{Mi} & \cdots & \Sigma_{MM} \end{bmatrix} \in \mathcal{R}^{m \times m}.$$

Note that there may exist (process) couplings between the sub-systems. We assume

- if
$$\mathcal{E}\left(\varepsilon_i^T \varepsilon_j\right) = \Sigma_{ij} = \Sigma_{ji}^T \neq 0, i, j = 1, \cdots, M, i \neq j,$$
the nodes i, j are networked and thus $(i, j) \in E$. That is, (i, j) is an edge of the graph;
- no sub-system is totally decoupled from the other sub-systems, which also means, the network is connected.

Our task is to detect the faults in sub-systems on the assumption that they do not occur simultaneously. To be specific, it is formulated as: given

$$y = \mathcal{E}y + \varepsilon \in \mathcal{R}^m, \varepsilon \sim \mathcal{N}(0, \Sigma), \mathcal{E}y = \begin{cases} \bar{y}, \text{ fault-free,} \\ \bar{y} + \Xi_i f_i, \text{ faulty,} \end{cases} \tag{14.73}$$

$$\Xi_i = \begin{bmatrix} 0 \\ \vdots \\ I_{m_i \times m_i} \\ \vdots \\ 0 \end{bmatrix} \Longrightarrow \Xi_i f_i = \begin{bmatrix} 0 \\ \vdots \\ f_i \\ \vdots \\ 0 \end{bmatrix},$$

where $i \in \{1, \cdots, M\}$, $\bar{y}$, Σ are unknown, develop a data-driven distributed fault detection scheme and algorithm aiming at optimally detecting f_i. Here, the data-driven fashion means an identification of $\bar{y}$, Σ in a training phase using the recorded process data, while distributed handling requires that only communications between the neighbouring nodes, both in the training and online detection phases, are allowed.

In Sect. 3.2 and Sub-section 3.3.3, we have discussed about the possible solution and interpretations of the above fault detection problem, when data processing is performed in the centralised manner. We summarise the relevant results in the following theorem, which is also the starting point for our subsequent work.

Theorem 14.5 *Given the model (14.73), then in the fault-free case*

$$cov\left(\Xi_i^- y\right) = cov\left(y_i - \hat{y}_i\right), \tag{14.74}$$

where $\hat{y}_i$ is the LMS estimation of $(y_i - \mathcal{E}y_i)$ given by

$$\hat{y}_i = cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right)\left(y^i - \mathcal{E}y^i\right), \tag{14.75}$$

$$cov\left(y_i, y^i\right) = \left[\Sigma_{i1} \cdots \Sigma_{ii-1} \; \Sigma_{ii+1} \cdots \Sigma_{iM}\right], \tag{14.76}$$

$$cov\left(y^i\right) = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1i-1} & \Sigma_{1i+1} & \cdots & \Sigma_{1M} \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ \Sigma_{i-11} & \cdots & \Sigma_{i-1i-1} & \Sigma_{i-1i+1} & \cdots & \Sigma_{i-1M} \\ \Sigma_{i+11} & \cdots & \Sigma_{i+1i-1} & \Sigma_{i+1i+1} & \cdots & \Sigma_{i+1M} \\ \vdots & & \vdots & \vdots & & \vdots \\ \Sigma_{M1} & \cdots & \Sigma_{Mi-1} & \Sigma_{Mi+1} & \cdots & \Sigma_{MM} \end{bmatrix}, y^i = \begin{bmatrix} y_1 \\ \vdots \\ y_{i-1} \\ y_{i+1} \\ \vdots \\ y_M \end{bmatrix}, \tag{14.77}$$

and

$$\Xi_i^- = \left(\Xi_i^T \Sigma^{-1} \Xi_i\right)^{-1} \Xi_i^T \Sigma^{-1}.$$

Proof For the simplicity of the notation and proof, we first introduce a transformation matrix

$$T=\begin{bmatrix} 0 \cdots 0 \; I & 0 \\ I \; \ddots \; \vdots \; \vdots & \vdots \\ \vdots \; \ddots \; 0 \; 0 & \vdots \\ 0 \cdots I \; 0 & 0 \\ 0 \cdots 0 \; 0 & I_{\bar{m}_{i+1}\times \bar{m}_{i+1}} \end{bmatrix} \in \mathcal{R}^{m\times m}, \bar{m}_{i+1}=\sum_{j=i+1}^{M} m_j,$$

which transforms y to

$$Ty=\begin{bmatrix} y_i \\ y_1 \\ \vdots \\ y_{i-1} \\ y_{i+1} \\ \vdots \\ y_M \end{bmatrix}.$$

Moreover, it holds

$$TT^T=I_{m\times m}, \Xi_i^T T^T=\begin{bmatrix} I_{m_i\times m_i} & 0 \cdots 0 \end{bmatrix},$$

$$T\Sigma T^T=\begin{bmatrix} \Sigma_{ii} & \Sigma_{i1} & \cdots & \Sigma_{ii-1} & \Sigma_{ii+1} & \cdots & \Sigma_{iM} \\ \Sigma_{1i} & \Sigma_{11} & \cdots & \Sigma_{1i-1} & \Sigma_{1i+1} & \cdots & \Sigma_{1M} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \Sigma_{i-1i} & \Sigma_{i-11} & \cdots & \Sigma_{i-1i-1} & \Sigma_{i-1i+1} & \cdots & \Sigma_{i-1M} \\ \Sigma_{i+1i} & \Sigma_{i+11} & \cdots & \Sigma_{i+1i-1} & \Sigma_{i+1i+1} & \cdots & \Sigma_{i+1M} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Sigma_{Mi} & \Sigma_{M1} & \cdots & \Sigma_{Mi-1} & \Sigma_{Mi+1} & \cdots & \Sigma_{MM} \end{bmatrix}.$$

Note that

$$cov\left(\Xi_i^- y\right)=\left(\Xi_i^T\Sigma^{-1}\Xi_i\right)^{-1}=\left(\Xi_i^T T^T\left(T\Sigma T^T\right)^{-1}T\Xi_i\right)^{-1}.$$

According to the well-known rule for the inverse of block matrices,

$$\begin{bmatrix} A & D \\ C & B \end{bmatrix}^{-1}=\begin{bmatrix} \left(A-DB^{-1}C\right)^{-1} & X \\ X & X \end{bmatrix},$$

where only the first block is of interest for our study, we have

$$\Xi_i^T T^T\left(T\Sigma T^T\right)^{-1}T\Xi_i=\left(\Sigma_{ii}-cov\left(y_i,y^i\right)\left(cov\left(y^i\right)\right)^{-1}\left(cov\left(y_i,y^i\right)\right)^T\right)^{-1}.$$

On the other hand,

$$\begin{aligned} cov\left(y_i - \hat{y}_i\right) &= \Sigma_{ii} + cov\left(y_i, y^i\right)\left(cov\left(y^i\right)\right)^{-1}\left(cov\left(y_i, y^i\right)\right)^T \\ &\quad -2\mathcal{E}\left(y_i - \mathcal{E}y_i\right)\left(cov\left(y_i, y^i\right)\left(cov\left(y^i\right)\right)^{-1}\left(y^i - \mathcal{E}y^i\right)\right)^T \\ &= \Sigma_{ii} - cov\left(y_i, y^i\right)\left(cov\left(y^i\right)\right)^{-1}\left(cov\left(y_i, y^i\right)\right)^T. \end{aligned}$$

Hence, (14.74) is proved.

It follows from this theorem that correlations among the sub-systems can be utilised to reduce the uncertainty (in the context of variance) and thus to enhance the fault detectability. To this end, a correlation-based estimation of the variable under consideration offers an optimal solution. In the next sub-sections, we will investigate the distributed realisation of a correlation-based fault detection algorithm. The main task is the development of

- a distributed iteration (learning) algorithm to determine

$$cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right),$$

which is called regression model and is needed both for the LMS estimate $\hat{y}_i$ as well as covariance matrix computation,

$$\Sigma_i = \Sigma_{ii} - cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right)\left(cov\left(y_i, y^i\right)\right)^T,$$

at the i-th node, and

- an online distributed algorithm based on the regression model for computing $\hat{y}_i$ at the i-th node.

The major challenge in dealing with these issues is the computation of $cov^{-1}\left(y^i\right)$, which should be, due to the high dimension of the process and its distributed communication topology, realised in a distributed fashion.

14.6.2 A Basic Iteration Learning Algorithm

For our purpose, we first formulate our task as follows: Given $cov\left(y_i, y^i\right)$ and $cov\left(y^i\right)$, as defined in (14.76)–(14.77), find a distributed iterative algorithm for the computation of the regression model

$$cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right) =: \left(Z^i\right)^T \in \mathcal{R}^{m_i \times (m-m_i)}. \tag{14.78}$$

Note that we can re-write (14.78) as

$$cov\left(y^{i}\right)Z^{i}=\left(cov\left(y_{i},y^{i}\right)\right)^{T}=\begin{bmatrix}\Sigma_{i1}^{T}\\ \vdots\\ \Sigma_{ii-1}^{T}\\ \Sigma_{ii+1}^{T}\\ \vdots\\ \Sigma_{iM}^{T}\end{bmatrix}. \tag{14.79}$$

Thus, our task can be formulated as solving linear equation (14.79) for Z^i using a numerical method. To this end, we adopt the well-known Richardson's method, which is widely applied for the computation of an iterative solution of a (high-dimensional) linear equation.

A general class of Richardson's iterations can be written as

$$Z_{k+1}^{i}=Z_{k}^{i}+\lambda\left(\left(cov\left(y_{i},y^{i}\right)\right)^{T}-cov\left(y^{i}\right)Z_{k}^{i}\right) \tag{14.80}$$

with k as the iteration number and λ being a constant to be designed to guarantee the iteration convergence. The following result is known in the literature.

Proposition 14.1 If the iteration (14.80) converges, then Z_k^i converges to the solution of

$$cov\left(y^{i}\right)Z^{i}=\left(cov\left(y_{i},y^{i}\right)\right)^{T}.$$

Next, we briefly study the conditions for selecting λ to guarantee the iteration convergence. Let

$$E_k=Z_k^i-Z^i$$

and re-write (14.80) into

$$\begin{aligned}E_{k+1}&=E_k+\lambda\left(cov\left(y^{i}\right)Z^{i}-cov\left(y^{i}\right)Z_{k}^{i}\right)\\&=\left(I-\lambda cov\left(y^{i}\right)\right)E_k.\end{aligned} \tag{14.81}$$

It is well-known that the convergence of the iteration (14.81) is equivalent to the stability of a discrete-time dynamic system defined by (14.81). Consequently, (14.81) converges, when $I-\lambda cov\left(y^{i}\right)$ is a Schur matrix. That means, all eigenvalues of $I-\lambda cov\left(y^{i}\right)$ are located inside the unit disk. Although such a λ generally exists and can be, considering $cov\left(y^{i}\right)$ being regular, determined in different ways, it becomes a challenging issue, when the iteration algorithm should run in a distributed fashion. This will be addressed in the sequel.

For our purpose, we write Z^i into

$$Z^i = \begin{bmatrix} Z_1 \\ \vdots \\ Z_{i-1} \\ Z_{i+1} \\ \vdots \\ Z_M \end{bmatrix}, Z_j \in \mathcal{R}^{m_j \times m_i}, j = 1, \cdots, M, j \neq i.$$

A distributed realisation of iteration algorithm (14.80) is understood as

- at the j-th node, $Z_j, j \in \{1, \cdots, M\}, j \neq i$, is computed, and
- for the computation of Z_j only $cov\left(y_j, y_l\right), l \in \mathcal{N}_j$, are available.

The distributed iteration can then be formulated as

$$Z_{j,k+1} = Z_{j,k} + \lambda \left(\Sigma_{ij}^T - \sum_{l \in \mathcal{N}_j} \Sigma_{jl} Z_{l,k} \right), j \in \{1, \cdots, M\}, j \neq i. \tag{14.82}$$

Note that

$$\Sigma_{jl} = cov\left(y_j, y_l\right), l \in \mathcal{N}_j,$$

denotes the correlation between the random variables of the j-th node and the ones of its neighbouring nodes. The key for a successful implementation of the iteration (14.82) is the cooperative determination of λ so that the iteration converges. Although there exist a number of algorithms for a distributed selection and even optimisation of λ, which is then iteration-depending (and thus varying with iterations), they are often used for online computation and thus strongly focused on a maximal convergence rate. Recall that our task of solving (14.78) is a part of the training for determining the needed parameters for the online estimation of y_i, when the process under consideration is in operation. Hence, there is no high real-time requirement. Instead, less involved computations are of considerable practical interest. Motivated by this, we propose below an algorithm for determining λ by means of a cooperation among the nodes.

The fact that $cov\left(y^i\right)$ is symmetric and positive definite ensures that all eigenvalues of $cov\left(y^i\right)$ are real and positive. In fact, $cov\left(y^i\right)$ can be written, by an SVD, as

$$\begin{gathered} cov\left(y^i\right) = U \Lambda U^T, U \in \mathcal{R}^{(m-m_i)\times(m-m_i)}, UU^T = I, U^T = U^{-1}, \\ \Lambda = diag\left(\lambda_1, \cdots, \lambda_{m-m_i}\right), \lambda_j > 0, j = 1, \cdots, m - m_i, \end{gathered}$$

with λ_j being an eigenvalue (singular value) of $cov\left(y^i\right)$. As a result,

$$I - \lambda cov\left(y^i\right) = U diag\left(1 - \lambda\lambda_1, \cdots, 1 - \lambda\lambda_{m-m_i}\right) U^{-1}. \tag{14.83}$$

It is clear from (14.83) that for

$$0 < \lambda < \frac{2}{\lambda_{\max}\left(cov\left(y^i\right)\right)}, \tag{14.84}$$

all eigenvalues of $\left(I - \lambda cov\left(y^i\right)\right)$ will be located inside the unit disk. Here,

$$\lambda_{\max}\left(cov\left(y^i\right)\right) = \max_j \lambda_j,\ j = 1, \cdots, m - m_i.$$

Recall that for (symmetric) positive definite $cov\left(y^i\right)$,

$$\lambda_{\max}\left(cov\left(y^i\right)\right) = \left\|cov\left(y^i\right)\right\|_2 \leq \left\|cov\left(y^i\right)\right\|_\infty = \left\|cov\left(y^i\right)\right\|_1. \tag{14.85}$$

And moreover

$$\left\|cov\left(y^i\right)\right\|_\infty = \max_{1 \leq l \leq m - m_i} \sum_{j=1}^{m-m_i} \left|a_{lj}\right|, \tag{14.86}$$

where

$$\begin{aligned} cov\left(y^i\right) &= \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1i-1} & \Sigma_{1i+1} & \cdots & \Sigma_{1M} \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ \Sigma_{i-11} & \cdots & \Sigma_{i-1i-1} & \Sigma_{i-1i+1} & \cdots & \Sigma_{i-1M} \\ \Sigma_{i+11} & \cdots & \Sigma_{i+1i-1} & \Sigma_{i+1i+1} & \cdots & \Sigma_{i+1M} \\ \vdots & & \vdots & \vdots & & \vdots \\ \Sigma_{M1} & \cdots & \Sigma_{Mi-1} & \Sigma_{Mi+1} & \cdots & \Sigma_{MM} \end{bmatrix} \\ &=: \left(a_{lj}\right), l, j = 1, \cdots, m - m_i. \end{aligned}$$

As a result, we can set

$$\lambda = \frac{2}{\left\|cov\left(y^i\right)\right\|_\infty + \epsilon} < \frac{2}{\lambda_{\max}\left(cov\left(y^i\right)\right)}$$

with $\epsilon > 0$ but sufficiently small, as far as $\left\|cov\left(y^i\right)\right\|_\infty$ is known.

Next, we propose a consensus algorithm of determining $\left\|cov\left(y^i\right)\right\|_\infty$ and setting of λ. The outputs of this algorithm are $\left\|cov\left(y^i\right)\right\|_\infty$ and λ being available at each node. It is assumed that d is the diameter of the graph of the sub-network with nodes $q = 2, \cdots, M$, and d is known.

Below is the algorithm. For the simplicity of notation, let $i = 1$ and suppose that $\epsilon > 0$ is a given.

Algorithm 14.8 *Distributed determination of* $\left\|cov\left(y^i\right)\right\|_\infty$

Step 0 *At the* q*-th node,* $q = 2, \cdots, M$*, set* $k = 0$ *and compute*

$$\gamma_{q,k} = \max_{\sum_{p=1}^{q-1} m_p+1 \leq l \leq \sum_{p=1}^{q} m_p} \sum_{j=1}^{m-m_i} \left| a_{lj} \right|; \tag{14.87}$$

Step 1 *The q-th node, $q = 2, \cdots, M$, communicates with its neighbours, including sending $\gamma_{q,k}$ to node $r, r \in \mathcal{N}_q$ and receiving $\gamma_{r,k}, r \in \mathcal{N}_q$;*

Step 2 *Compute*

$$\gamma_{q,k+1} = \max \left\{ \gamma_{q,k}, \gamma_{r,k}, r \in \mathcal{N}_q \right\}, \tag{14.88}$$

set $k = k + 1$; If $k < d$

Step 3 *Go to Step 1*

Step 4 *Otherwise, for $k = d$, set*

$$\lambda = \frac{2}{\gamma_{q,k} + \epsilon}. \tag{14.89}$$

Theorem 14.6 *After d iterations,*

$$\gamma_{q,k} = \left\| cov\left(y^i\right) \right\|_\infty, q = 2, \cdots, M.$$

Proof The proof is evident. According to (14.87), after the computations in Step 0, at least at one node, say node j,

$$\gamma_{j,0} = \left\| cov\left(y^i\right) \right\|_\infty.$$

Since the greatest distance between the j-node and any other nodes is equal to d, thanks to the iterative rule (14.88), $\gamma_{j,0}$ should reach all nodes after d iterations.

Having determined λ, the distributed recursion (14.80) or equivalently (14.82) can be activated and runs until it converges, as the tolerance

$$\left\| Z_{j,k} - Z_j \right\| \leq \gamma$$

is reached. At the end of this learning procedure,

$$Z_j := Z_{j,k}$$

is saved in the j-th node, $j \in \{1, \cdots, M\}$, $j \neq i$. Moreover, the node $j \in \mathcal{N}_i$ sends Z_j to the i-th node so that

$$\begin{aligned} \varSigma_i &= \varSigma_{ii} - cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right) \left(cov\left(y_i, y^i\right)\right)^T \\ &= \varSigma_{ii} - \sum_{j \in \mathcal{N}_i} \varSigma_{ij} Z_j \end{aligned}$$

and further Σ_i^{-1} can be computed at the i-th node, which is needed for the online computation of the T^2 test statistic

$$J_{T_i^2} = \left(y_i - \hat{y}_i\right)^T \Sigma_i^{-1} \left(y_i - \hat{y}_i\right). \tag{14.90}$$

Note that $\hat{y}_i$ is the LMS estimation of $(y_i - \mathcal{E}y_i)$, as defined in Theorem 14.5. Finally, the corresponding threshold is

$$J_{th,i} = \chi_\alpha^2 (m_i).$$

Here, it is assumed that sufficient number of data are collected.

14.6.3 Online Detection Algorithm

Recall that the online detection consists of

- computation of $J_{T_i^2}$,
- decision making

$$\begin{cases} J_{T_i^2} - J_{th,i} \le 0 \Longrightarrow \text{fault-free}, \\ J_{T_i^2} - J_{th,i} > 0 \Longrightarrow \text{faulty}. \end{cases}$$

The computation of $\hat{y}_i$,

$$\begin{aligned} \hat{y}_i &= cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right)\left(y^i - \mathcal{E}y^i\right) \\ &= \left(Z^i\right)^T \left(y^i - \mathcal{E}y^i\right) = \sum_{j \in \{1, \cdots, M, j \neq i\}} Z_j^T \left(y_j - \mathcal{E}y_j\right), \end{aligned} \tag{14.91}$$

requires the transmission of $Z_j^T \left(y_j - \mathcal{E}y_j\right)$ from the j-th node to the i-th node, $j \in \{1, \cdots, M, j \neq i\}$. For this reason, a transmission protocol will be designed. It should be remarked that the data transmission from those nodes, which are located far from the i-th node (in the context of greater distance), takes time and could be remarkably corrupted with noises of the communication channels. On the other hand, the correlation of the process variables at these nodes with the ones of the i-th node is in general (very) weak. In fact, with the increasing of the distance, the correlation will become weaker, as will be shown in the subsequent section.

14.7 Analysis and Alternative Algorithms

In the last section, we have learnt that correlation-based estimation of (local) process variables is useful to reduce the noise-induced uncertainty and hence improve the fault detection performance. To this end, the computation of $\hat{y}_i$, which should run in the i-th node as given in (14.91), plays a central role. Since data transmissions are needed for being able to run (14.91) in the i-th node, it is of interest to know the relation between the distance from the j-th node to the i-th node and its contribution to the estimate $\hat{y}_i$. The motivation for this question is the fact that, due to the channel noises, the transmitted data will be strongly corrupted with noises as the distance increases. In this section, we will first try to answer the raised question and then propose some alternative algorithms.

14.7.1 Couplings, Correlations and Estimation

Recall

$$\hat{y}_i = cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right)\left(y^i - \mathcal{E}y^i\right).$$

Here,

$$cov\left(y_i, y^i\right) = \left[\, \Sigma_{i1} \cdots \Sigma_{ii-1} \;\; \Sigma_{ii+1} \cdots \Sigma_{iM} \,\right]$$

represents the correlations between the i-th sub-system and the remaining sub-systems. If

$$\Sigma_{ij} \neq 0, j \in \{1, \cdots, M, j \neq i\},$$

we say, from the system point of view, there exists a coupling between the i-th and the j-th sub-systems, or from the statistic point of view, the i-th and j-th sub-systems are correlated. It is worth mentioning that large-scale distributed systems are often characterised by their weak couplings between the sub-systems. Remember our assumption that the i-th and j-th nodes are connected as far as $\Sigma_{ij} \neq 0$. Consequently, for a large-scale system, most sub-matrices in $cov\left(y_i, y^i\right)$ are zero. The corresponding communication network is said to be sparse. In fact, the sparseness of the system and network configurations is the further motivation for our subsequent study.

We now consider the computation of $cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right)$. For our purpose, we first define

$$\mathcal{Y}_{\mathcal{N}_i} = \left\{y_j, j \neq i, j \in \mathcal{N}_i\right\}, \mathcal{Y}_{\bar{\mathcal{N}}_i} = \left\{y_j, j \neq i, j \notin \mathcal{N}_i\right\},$$

and order the vectors in $\mathcal{Y}_{\mathcal{N}_i}, \mathcal{Y}_{\bar{\mathcal{N}}_i}$ into two vectors, respectively,

$$y_{\mathcal{N}_i} = \begin{bmatrix} \vdots \\ y_j \\ \vdots \end{bmatrix}, y_j \in \mathcal{Y}_{\mathcal{N}_i}, y_{\bar{\mathcal{N}}_i} = \begin{bmatrix} \vdots \\ y_l \\ \vdots \end{bmatrix}, y_l \in \mathcal{Y}_{\bar{\mathcal{N}}_i}.$$

Without loss of generality, it is assumed that

$$y^i = \begin{bmatrix} y_1 \\ \vdots \\ y_{i-1} \\ y_{i+1} \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} y_{\mathcal{N}_i} \\ y_{\bar{\mathcal{N}}_i} \end{bmatrix} \tag{14.92}$$

and moreover, for the simplification of the description, $y_{\mathcal{N}_i}, y_{\bar{\mathcal{N}}_i}$ are centered (zero mean).

Remark 14.8 *We would like to emphasise that the centralisation of y^i can be done at each node and thus in a distributed manner. Hence, the assumption*

$$\mathcal{E} y^i = 0$$

does not affect the generality of our subsequent study. It is made only for the purpose of simplifying notation.

Now, $y_{\mathcal{N}_i}$ includes all measurement vectors of the sub-systems which are correlated with the i-th sub-system. In against, the measurement vectors included in $y_{\bar{\mathcal{N}}_i}$ are uncorrelated with y_i. Corresponding to $y_{\mathcal{N}_i}, y_{\bar{\mathcal{N}}_i}$, we denote

$$cov\left(y^i\right) = \begin{bmatrix} \Sigma_{\mathcal{N}_i,\mathcal{N}_i} & \Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i} \\ \Sigma_{\bar{\mathcal{N}}_i,\mathcal{N}_i} & \Sigma_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i} \end{bmatrix}, cov\left(y_i, y^i\right) = \begin{bmatrix} \Sigma_{i,\mathcal{N}_i} & 0 \end{bmatrix}.$$

Applying the well-known formula for the inverse computation of 2 × 2 block matrices to $cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right)$ results in

$$cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right) = \begin{bmatrix} \Sigma_{i,\mathcal{N}_i} \Delta^{-1} & -\Sigma_{i,\mathcal{N}_i} \Delta^{-1} \Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i} \Sigma^{-1}_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i} \end{bmatrix},$$
$$\Delta = \Sigma_{\mathcal{N}_i,\mathcal{N}_i} - \Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i} \Sigma^{-1}_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i} \Sigma_{\bar{\mathcal{N}}_i,\mathcal{N}_i}.$$

This allows us to write $\hat{y}_i$ as

$$\hat{y}_i = \Sigma_{i,\mathcal{N}_i} \Delta^{-1} \left(y_{\mathcal{N}_i} - \Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i} \Sigma^{-1}_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i} y_{\bar{\mathcal{N}}_i} \right). \tag{14.93}$$

The alternative computation form of $\hat{y}_i$ given in (14.93) is of fundamental significance for our subsequent work and worth for us to gain a deep insight into it. It is clear that $\Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i}\Sigma^{-1}_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i}y_{\bar{\mathcal{N}}_i}$ is an LMS estimate for $y_{\mathcal{N}_i}$ using the data vector $y_{\bar{\mathcal{N}}_i}$. Note that $y_{\bar{\mathcal{N}}_i}$ is composed of all the measurement vectors belonging to $\mathcal{Y}_{\bar{\mathcal{N}}_i}$. That is, they are uncorrelated with y_i. Moreover, notice the fact that

$$\Delta = \Sigma_{\mathcal{N}_i,\mathcal{N}_i} - \Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i}\Sigma^{-1}_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i}\Sigma_{\bar{\mathcal{N}}_i,\mathcal{N}_i}$$

is the covariance matrix of

$$\Delta y_{\mathcal{N}_i} := y_{\mathcal{N}_i} - \Sigma_{\mathcal{N}_i,\bar{\mathcal{N}}_i}\Sigma^{-1}_{\bar{\mathcal{N}}_i,\bar{\mathcal{N}}_i}y_{\bar{\mathcal{N}}_i}, \tag{14.94}$$

which is minimum. As a result, $\hat{y}_i$ given in (14.93) can be interpreted as an LMS estimate of y_i based on the data $\Delta y_{\mathcal{N}_i}$ received from its neighbours. $\Delta y_{\mathcal{N}_i}$ is the difference between the measurement vectors at the nodes in $\mathcal{N}_i$ and their estimate using the measurement vectors in $\mathcal{Y}_{\bar{\mathcal{N}}_i}$, and is of minimum variance. We would like to call reader's attention to the following two aspects from this interpretation.

Firstly, the problem of the LMS estimation of y_i is reduced to the LMS estimation of $y_{\mathcal{N}_i}$ using the data from the nodes which are not (directly) connected to the i-th node. Similarly, the LMS estimation of $y_{\mathcal{N}_i}$ can be further reduced to the LMS estimation of those measurement vectors belonging to $\mathcal{Y}_{\bar{\mathcal{N}}_i}$ but being correlated with $y_{\mathcal{N}_i}$. Continuing this procedure, it is clear that the original LMS estimation problem can be decomposed into a (finite) number of embedded LMS estimation sub-problems, which can be written in a recursive form. To describe it in detail, we introduce the following notations:

- $\mathcal{N}_i^k, k = 0, \cdots, d$, denotes the set of those nodes in the graph whose distance to the i-th node equals to k,

  $$\mathcal{N}_i^k = \{j, d(i, j) = k\}, \mathcal{N}_i^0 = i, \mathcal{N}_i^1 = \mathcal{N}_i,$$

  where $d(i, j)$ denotes the distance between the node i and node j, d is the diameter of the graph, as defined in (14.3). Note that $\mathcal{N}_i^k$ can also be computed recursively

  $$\mathcal{N}_i^k = \left\{j, j \in \mathcal{N}_l, l \in \mathcal{N}_i^{k-1}\right\};$$

- $\bar{\mathcal{N}}_i^k, k = 0, \cdots, d-1$, is defined by

  $$\bar{\mathcal{N}}_i^k = \{j, d(i, j) > k\};$$

- $y_{\mathcal{N}_i^{k+1}}, y_{\bar{\mathcal{N}}_i^k}, k = 0, \cdots, d-1$, consist of all measurement vectors of the nodes belonging to $\mathcal{N}_i^{k+1}, \bar{\mathcal{N}}_i^k$, respectively, and are thus given by

$$y_{\mathcal{N}_i^{k+1}} = \begin{bmatrix} \vdots \\ y_j \\ \vdots \end{bmatrix}, j \in \mathcal{N}_i^{k+1}, y_{\bar{\mathcal{N}}_i^k} = \begin{bmatrix} \vdots \\ y_l \\ \vdots \end{bmatrix}, l \in \bar{\mathcal{N}}_i^k;$$

- $\Sigma_{\mathcal{N}_i^l,\mathcal{N}_i^\gamma}$, $\Sigma_{\bar{\mathcal{N}}_i^l,\mathcal{N}_i^\gamma}$, $\Sigma_{\bar{\mathcal{N}}_i^l,\bar{\mathcal{N}}_i^\gamma}$ are the covariance matrices defined, respectively, by

$$\Sigma_{\mathcal{N}_i^l,\mathcal{N}_i^\gamma} = cov\left(y_{\mathcal{N}_i^l}, y_{\mathcal{N}_i^\gamma}\right), \Sigma_{\bar{\mathcal{N}}_i^l,\mathcal{N}_i^\gamma} = cov\left(y_{\bar{\mathcal{N}}_i^l}, y_{\mathcal{N}_i^\gamma}\right),$$
$$\Sigma_{\bar{\mathcal{N}}_i^l,\bar{\mathcal{N}}_i^\gamma} = cov\left(y_{\bar{\mathcal{N}}_i^l}, y_{\bar{\mathcal{N}}_i^\gamma}\right),$$

for some integer l, γ.

With the aid of these notations, $\hat{y}_i$ given in (14.93) can be (backwards) recursively computed as follows:

$$\begin{aligned} \hat{y}_{\mathcal{N}_i^d} &= 0, \\ \hat{y}_{\mathcal{N}_i^k} &= \Sigma_{\mathcal{N}_i^k,\mathcal{N}_i^{k+1}} \Delta_{k+1}^{-1} \left(y_{\mathcal{N}_i^{k+1}} - \hat{y}_{\mathcal{N}_i^{k+1}}\right), k = 0, \cdots, d-1, \\ \Delta_{k+1} &= cov\left(y_{\mathcal{N}_i^{k+1}} - \hat{y}_{\mathcal{N}_i^{k+1}}\right) \\ &= \Sigma_{\mathcal{N}_i^{k+1},\mathcal{N}_i^{k+1}} - \Sigma_{\mathcal{N}_i^{k+1},\bar{\mathcal{N}}_i^{k+1}} \Sigma^{-1}_{\bar{\mathcal{N}}_i^{k+1},\bar{\mathcal{N}}_i^{k+1}} \Sigma_{\bar{\mathcal{N}}_i^{k+1},\mathcal{N}_i^{k+1}}, \\ \hat{y}_i &= \hat{y}_{\mathcal{N}_i^0} = \Sigma_{\mathcal{N}_i^0,\mathcal{N}_i^1} \Delta_1^{-1} \left(y_{\mathcal{N}_i^1} - \hat{y}_{\mathcal{N}_i^1}\right), \end{aligned} \tag{14.95}$$

whose core is the estimation computation (14.95).

Secondly, the above recursive algorithm reveals that the contribution of the measurement vector of the j-th node to the uncertainty reduction at the node i depends on the distance between these two nodes. In fact, with the increasing distance, this contribution will become weaker. To see this clearly, recall

$$\hat{y}_i = cov\left(y_i, y^i\right) cov^{-1}\left(y^i\right) y^i = \left(Z^i\right)^T y^i = \sum_{j \in \{1,\cdots,M, j \neq i\}} Z_j^T y_j.$$

Let

$$Z_{\mathcal{N}_i^l}^T = \left[\cdots Z_j^T \cdots\right], j \in \mathcal{N}_i^l$$

be the mapping matrix from $y_{\mathcal{N}_i^l}$ to $\hat{y}_i$, which consists of the mapping (weighting) sub-matrices from the measurement vectors of those nodes, whose distance to node i is l, $1 \leq l \leq d$, to the estimate $\hat{y}_i$. It follows from (14.95) that

$$Z_{\mathcal{N}_i^l}^T = \prod_{k=0}^{l-1} \Sigma_{\mathcal{N}_i^k,\mathcal{N}_i^{k+1}} \Delta_{k+1}^{-1}.$$

Since Δ_{k+1}^{-1} serves for a normalisation of the process data $y_{\mathcal{N}_i^{k+1}} - \hat{y}_{\mathcal{N}_i^{k+1}}$,

$$\bar{\Sigma}_{\mathcal{N}_i^k,\mathcal{N}_i^{k+1}} := \Sigma_{\mathcal{N}_i^k,\mathcal{N}_i^{k+1}} \Delta_{k+1}^{-1}$$

can be viewed as the normalised correlation between $y_{\mathcal{N}_i^{k+1}}, y_{\mathcal{N}_i^k}$. In other words, the mapping from $y_{\mathcal{N}_i^l}$ to $\hat{y}_i$ is exactly the product of the normalised correlations and so the couplings

$$Z_{\mathcal{N}_i^l}^T = \bar{\Sigma}_{\mathcal{N}_i^0,\mathcal{N}_i^1} \cdots \bar{\Sigma}_{\mathcal{N}_i^{l-1},\mathcal{N}_i^l}. \tag{14.96}$$

Because the process under consideration is weakly coupled, it is evident that for a larger l, the contribution (via the weighting matrix $Z_{\mathcal{N}_i^l}^T$) of $y_{\mathcal{N}_i^l}$ to the estimation of y_i and so that to the uncertainty reduction at the node i is weaker.

14.7.2 Alternative Algorithms

The discussions and the results given in (14.95)–(14.96) motivate us

- to modify the correlation-based fault detection schemes algorithm proposed in the last section, and
- to propose alternative algorithms.

A modified algorithm

Consider the iterative algorithm proposed in Sub-section 14.6.2. On the assumption that the sub-systems are weakly coupled, it is reasonable to neglect the (weaker) contributions of those sub-systems to $\hat{y}_i$, which are located far away from the i-th node. Let $l << d$ be the maximal distance from the i-th node to those nodes, which are taken into account for estimating y_i. We define

$$\mathcal{N}_i^{\leq l} = \bigcup_{k=1}^{l} \mathcal{N}_i^k$$

as the set of all nodes whose distance to the i-th node is not greater than l, and

$$y_{\mathcal{N}_i^{\leq l}} = \begin{bmatrix} \vdots \\ y_j \\ \vdots \end{bmatrix}, j \in \mathcal{N}_i^{\leq l}$$

as the vector that consists of all the measurement vectors from the nodes in $\mathcal{N}_i^{\leq l}$. For our purpose, the problem to be solved is now formulated as: Given

$$cov\left(y_i, y_{\mathcal{N}_i^{\leq l}}\right) = \begin{bmatrix} \cdots & \Sigma_{ij} & \cdots \end{bmatrix}, j \in \mathcal{N}_i^{\leq l},$$

$$cov\left(y_{\mathcal{N}_i^{\leq l}}\right) = \begin{bmatrix} \ddots & \vdots & \vdots \\ \cdots & \Sigma_{lj} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, l, j \in \mathcal{N}_i^{\leq l},$$

find $\left(Z_{\mathcal{N}_i^{\leq l}}\right)^T$ defined by

$$\left(Z_{\mathcal{N}_i^{\leq l}}\right)^T = cov\left(y_i, y_{\mathcal{N}_i^{\leq l}}\right) cov^{-1}\left(Z_{\mathcal{N}_i^{\leq l}}\right) \Longleftrightarrow$$

$$cov\left(y_{\mathcal{N}_i^{\leq l}}\right)\left(Z_{\mathcal{N}_i^{\leq l}}\right)^T = cov\left(y_i, y_{\mathcal{N}_i^{\leq l}}\right), \left(Z_{\mathcal{N}_i^{\leq l}}\right)^T = \begin{bmatrix} \vdots \\ Z_j^T \\ \vdots \end{bmatrix}, j \in \mathcal{N}_i^{\leq l} \quad (14.97)$$

in a distributed fashion. We adopt the recursive algorithm (14.82) for solving the problem as follows

$$Z_{j,k+1} = Z_{j,k} + \lambda\left(\Sigma_{ij}^T - \sum_{p \in \mathcal{N}_j^1} \Sigma_{jl} Z_{p,k}\right), j \in \mathcal{N}_i^{\leq l}, \quad (14.98)$$

where $Z_{j,k}$ denotes the computed value of matrix Z_j at the k-th iteration with Z_j being a sub-matrix in $Z_{\mathcal{N}_i^{\leq l}}$. λ is determined by running the consensus algorithm given in the last section and using (14.89),

$$\lambda = \frac{2}{\gamma_{q,l} + \epsilon}, q \in \mathcal{N}_i^{\leq l}. \quad (14.99)$$

Since the maximal distance between any two nodes in $\mathcal{N}_i^{\leq l}$ is not greater than l, it follows from Theorem 14.6 that after l iterations

$$\gamma_{q,l} = \left\| cov\left(y_{\mathcal{N}_i^{\leq l}}\right) \right\|_\infty, q \in \mathcal{N}_i^{\leq l}.$$

As illustrated in the last section, by means of λ given in (14.99), it holds

$$\lim_{k \to \infty} Z_{j,k} = Z_j.$$

Once $Z_{\mathcal{N}_i^{\leq l}}$ is determined,

$$\begin{aligned}\hat{y}_i &= cov\left(y_i, y_{\mathcal{N}_i^{\leq l}}\right) cov^{-1}\left(y_{\mathcal{N}_i^{\leq l}}\right) y_{\mathcal{N}_i^{\leq l}} = \left(Z_{\mathcal{N}_i^{\leq l}}\right)^T y_{\mathcal{N}_i^{\leq l}}, \qquad (14.100)\\ \varSigma_i &= \varSigma_{ii} - cov\left(y_i, y_{\mathcal{N}_i^{\leq l}}\right) cov^{-1}\left(y_{\mathcal{N}_i^{\leq l}}\right)\left(cov\left(y_i, y_{\mathcal{N}_i^{\leq l}}\right)\right)^T\\ &= \varSigma_{ii} - \sum_{j\in\mathcal{N}_i} \varSigma_{ij} Z_j\end{aligned}$$

can be online and distributed computed for the fault detection purpose.

Remark 14.9 *It is worth pointing out that $\hat{y}_i$ given in (14.100) is the LMS estimate of y_i using all the measurement vectors available at the node set $\mathcal{N}_i^{\leq l}$.*

An alternative algorithm

It is state of the art in the parallel computation area that a computation problem is decomposed into a number of sub-problems, which are then solved in parallel. For the computation of the inverse of a (high-dimensional) matrix, there are well-established algorithms performed following this strategy. In fact, the recursive form (14.95) suggests to realise the estimation in such a manner. Below, we propose an alternative algorithm that allows us

- similar to the previous algorithm, to estimate y_i using the measurement vectors available at the node set $\mathcal{N}_i^{\leq l}$ for some $l << d$, and
- to perform the training (learning) in finite steps instead of a recursive algorithm, as adopted in (14.82) or (14.98).

Given $l(<< d)$, we first investigate which computations should be performed at a node belonging to $\mathcal{N}_i^{\leq l}$, in order to achieve the LMS estimate of y_i. To this end, consider (14.95), which can be re-formed as

$$\begin{aligned}\hat{y}_{j_{\mathcal{N}_i^k}} &= \varSigma_{j,\mathcal{N}_i^{k+1}} \varDelta_{k+1}^{-1}\left(y_{\mathcal{N}_i^{k+1}} - \hat{y}_{\mathcal{N}_i^{k+1}}\right), \qquad (14.101)\\ \varSigma_{j,\mathcal{N}_i^{k+1}} &= \mathcal{E} y_{j_{\mathcal{N}_i^k}} y_{\mathcal{N}_i^{k+1}}^T, \; j_{\mathcal{N}_i^k} \in \mathcal{N}_i^k\end{aligned}$$

with $j_{\mathcal{N}_i^k}$ denoting the node under consideration. It is obvious that the computation of $\hat{y}_{j_{\mathcal{N}_i^k}}$ depends on the topology of the node $j_{\mathcal{N}_i^k}$ and its neighbours. For this reason, we check the possible configurations.

- When $j_{\mathcal{N}_i^k}$ is uncorrelated with any node in $\mathcal{N}_i^{k+1}$ or $k = l$, it holds

$$\hat{y}_{j_{\mathcal{N}_i^k}} = 0. \qquad (14.102)$$

- When $j_{\mathcal{N}_i^k}$ is correlated (and so connected) with some nodes in $\mathcal{N}_i^{k+1}$, which build set

$$\mathcal{N}_{j_{\mathcal{N}_i^k}} \subset \mathcal{N}_i^{k+1},$$

we further distinguish the following two sub-sets:

$$\begin{aligned} \mathcal{N}_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} &= \left\{ j, j \in \mathcal{N}_l \subset \mathcal{N}_i^{k+1}, l \in \mathcal{N}_{j_{\mathcal{N}_i^k}} \right\}, \\ \bar{\mathcal{N}}_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} &= \left\{ j, j \in \mathcal{N}_i^{k+1}, j \notin \mathcal{N}_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} \right\}. \end{aligned}$$

If we connect all nodes in $\mathcal{N}_i^{k+1}$ during the training, $\bar{\mathcal{N}}_{\mathcal{N}_{j_{\mathcal{N}_i^k}}}$ is empty and thus

$$\mathcal{N}_i^{k+1} = \mathcal{N}_{j_{\mathcal{N}_i^k}} \cup \bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}, \bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}} = \left\{ j, j \in \mathcal{N}_i^{k+1}, j \notin \mathcal{N}_{j_{\mathcal{N}_i^k}} \right\}.$$

Let

$$\Delta y_{\mathcal{N}_i^{k+1}} = y_{\mathcal{N}_i^{k+1}} - \hat{y}_{\mathcal{N}_i^{k+1}}$$

and split it into

$$\Delta y_{\mathcal{N}_i^{k+1}} = \begin{bmatrix} \Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} \\ \Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}} \end{bmatrix}, \Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} = \begin{bmatrix} \vdots \\ \Delta y_l \\ \vdots \end{bmatrix}, l \in \mathcal{N}_{j_{\mathcal{N}_i^k}}.$$

As a result, we have, according to (14.93),

$$\begin{aligned} \hat{y}_{j_{\mathcal{N}_i^k}} =& \Sigma_{j,\mathcal{N}_{j_{\mathcal{N}_i^k}}} \bar{\Delta}_{j,k}^{-1} \left(\Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} - \Delta \hat{y}_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} \right), \qquad (14.103) \\ \Delta \hat{y}_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} =& cov \left(\Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}}, \Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}} \right) cov^{-1} \left(\Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}} \right) \Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}}, \\ \Sigma_{j,\mathcal{N}_{j_{\mathcal{N}_i^k}}} =& cov \left(y_{j_{\mathcal{N}_i^k}}, y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} \right), \\ \bar{\Delta}_{j,k} =& cov \left(\Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} \right) \\ & - cov \left(\Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}}, \Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}} \right) cov^{-1} \left(\Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}} \right) cov \left(\Delta y_{\bar{\mathcal{N}}_{j_{\mathcal{N}_i^k}}}, \Delta y_{\mathcal{N}_{j_{\mathcal{N}_i^k}}} \right). \end{aligned}$$

In summary, $\hat{y}_{j_{\mathcal{N}_i^k}}$ can be computed according to either (14.102) or (14.103), so far sufficient information about the topology of the node $j_{\mathcal{N}_i^k}$ and its neighbours is available. Below are two algorithms that will run during the training phase.

Algorithm 14.9 *Algorithm of building necessary information and communication topology:*

Step 0: *Node i initials the start: set $k = 0, l$;*
Step 1: *Form $\mathcal{N}_j$, for all $j \in \mathcal{N}_i^k$, and build*

$$\mathcal{N}_i^{k+1} = \bigcup_{j \in \mathcal{N}_i^k} \mathcal{N}_j$$

at each node in $\mathcal{N}_i^k$ by exchanging data among the nodes in $\mathcal{N}_i^k$ and transmit $\mathcal{N}_i^{k+1}$ to the nodes in $\mathcal{N}_j$ for all $j \in \mathcal{N}_i^k$;

Step 2: *Set $k = k + 1$, label the nodes in $\mathcal{N}_i^k$ by i_k and connect all nodes in $\mathcal{N}_i^k$;*

Step 3: *If $k = l$, stop, otherwise, go to Step 1.*

It follows from (14.103) that for the online estimate $\hat{y}$, $j \in \mathcal{N}_i^k, k = 0, 1, \cdots, l$, following matrices are needed:

$$\Sigma_{j,\mathcal{N}_j}, cov\left(\Delta y_{\mathcal{N}_j}, \Delta y_{\bar{\mathcal{N}}_j}\right), cov\left(\Delta y_{\bar{\mathcal{N}}_j}\right), \bar{\Delta}_{j,k}.$$

While $\Sigma_{j,\mathcal{N}_j}$ can be directly estimated using the process data at node j and the data received from the nodes in $\mathcal{N}_j$,

$$cov\left(\Delta y_{\mathcal{N}_j}, \Delta y_{\bar{\mathcal{N}}_j}\right) = \begin{bmatrix} \vdots \\ cov\left(\Delta y_l, \Delta y_{\bar{\mathcal{N}}_j}\right) \\ \vdots \end{bmatrix}, l \in \mathcal{N}_j,$$

$cov\left(\Delta y_{\bar{\mathcal{N}}_j}\right)$, $\bar{\Delta}_{j,k}$ should be identified during the training phase using sufficient data and by means of data transmissions. The following algorithm serves for this purpose.

Algorithm 14.10 *Algorithm to identify the matrices to perform (14.93): Input data and parameters: $l, \mathcal{N}_i^k, k = 0, 1, \cdots, l$, $\Sigma_{j,\mathcal{N}_j}$, $j \in \mathcal{N}_i^k$ and sufficient data collected at each node in $\mathcal{N}_i^k, k = 0, 1, \cdots, l$.*

Step 0: *Set $k = l$ and*

$$\xi_j = y_j, j \in \mathcal{N}_i^k,$$

collect sufficient data of ξ_j and form the data matrix

$$\Xi_{j,k} = \left[\cdots \ \xi_j \ \cdots\right];$$

Step 1: *Set $k = k - 1$;*

Step 2: *For $j \in \mathcal{N}_i^k$*

Step 2-1: *If*

$$\mathcal{N}_i^k = \mathcal{N}_i^{k+1},$$

set

$$\Delta\Xi_l = \Xi_{l,k+1}, l \in \mathcal{N}_j,$$

and transmit $\Delta\Xi_l, l \in \mathcal{N}_j$, to node j and go to Step 2-2. Otherwise, transmit $\Xi_{q,k+1}, q \in \mathcal{N}_i^{k+1}, q \notin \mathcal{N}_j$ to nodes $l \in \mathcal{N}_j$; At nodes $l \in \mathcal{N}_j$, order the data into

$$\varXi_{\bar{\mathcal{N}}_l} = \begin{bmatrix} \vdots \\ \varXi_{q,k+1} \\ \vdots \end{bmatrix}, q \in \mathcal{N}_i^{k+1}, q \notin \mathcal{N}_j$$

and build

$$\Sigma_{\bar{\mathcal{N}}_l} = \varXi_{\bar{\mathcal{N}}_l} \varXi_{\bar{\mathcal{N}}_l}^T;$$

In parallel, set

$$\varXi_l = \varXi_{l,k+1}, l \in \mathcal{N}_j;$$

Compute and save at nodes $l \in \mathcal{N}_j$

$$\Sigma_{l,\bar{\mathcal{N}}_l} = \varXi_l \varXi_{\bar{\mathcal{N}}_l} = cov\left(\Delta y_l, \Delta y_{\bar{\mathcal{N}}_j}\right), \Sigma_{\bar{\mathcal{N}}_l}^{-1} = cov^{-1}\left(\Delta y_{\bar{\mathcal{N}}_j}\right),$$

build by sufficient data

$$\Delta \varXi_l = \varXi_l - \Sigma_{l,\bar{\mathcal{N}}_l} \Sigma_{\bar{\mathcal{N}}_l}^{-1} \varXi_{\bar{\mathcal{N}}_l},$$

and transmit $\Delta \varXi_l, l \in \mathcal{N}_j$, *to node* j;

Step 2-2: *Collect sufficient data at node* j, *set*

$$\varXi_j = \begin{bmatrix} \cdots \ y_j \ \cdots \end{bmatrix};$$

Form at node j

$$\Delta \varXi_j = \begin{bmatrix} \vdots \\ \Delta \varXi_l \\ \vdots \end{bmatrix}, \Sigma_j = \Delta \varXi_j \Delta \varXi_j^T = \bar{\Lambda}_{j,k},$$

compute and save

$$\Psi_j = \Sigma_{j,\mathcal{N}_j} \Sigma_j^{-1}$$

and using sufficient data to build

$$\hat{\varXi}_j = \Psi_j \Delta \varXi_j$$

Build

$$\tilde{\varXi}_{j,k} = \varXi_j - \hat{\varXi}_j$$

End (of Step 2)

Step 3: *If* $k = 0$, *compute and output*

$$\frac{1}{N-1}\tilde{\Xi}_{j,k}\tilde{\Xi}_{j,k}^{T}=cov\left(y_i-\hat{y}_i\right),$$

stop, otherwise, go to Step 1, where N is the number of data for forming Ξ_j.

Once all needed matrices are identified and saved distributed, the following algorithm can run online for the detection purpose.

Algorithm 14.11 *Online fault detection:*

Step 0: Set $k = l$ and collect

$$\xi_{j,k}=y_j,\ j\in\mathcal{N}_i^k;$$

Step 1: Set $k = k - 1$;
Step 2: For $j \in \mathcal{N}_i^k$

Step 2-1: If

$$\mathcal{N}_i^k=\mathcal{N}_i^{k+1},$$

set

$$\Delta\xi_l=\xi_{l,k+1},\ l\in\mathcal{N}_j,$$

and transmit $\Delta\xi_l, l \in \mathcal{N}_j$, to node j and go to Step 2-2. Otherwise, transmit

$$\xi_q=\xi_{q,k+1},\ q\in\mathcal{N}_i^{k+1},\ q\notin\mathcal{N}_j$$

to nodes $l \in \mathcal{N}_j$; At nodes $l \in \mathcal{N}_j$, order the data into

$$\xi_{\bar{\mathcal{N}}_l}=\begin{bmatrix}\vdots\\ \xi_q\\ \vdots\end{bmatrix},\ q\in\mathcal{N}_i^{k+1},\ q\notin\mathcal{N}_j$$

In parallel, set

$$\xi_l=\xi_{l,k+1},\ l\in\mathcal{N}_j$$

and compute at nodes $l \in \mathcal{N}_j$

$$\Delta\xi_l=\xi_l-\Sigma_{l,\bar{\mathcal{N}}_l}\Sigma_{\bar{\mathcal{N}}_l}^{-1}\xi_{\bar{\mathcal{N}}_l}$$

transmit $\Delta\xi_l, l \in \mathcal{N}_j$, to node j;
Step 2-2: Collect data y_j at node j, form

$$\Delta\xi_j = \begin{bmatrix} \vdots \\ \Delta\xi_l \\ \vdots \end{bmatrix}, l \in \mathcal{N}_j$$

and compute

$$\xi_{j,k} = y_j - \Psi_j \Delta\xi_j;$$

End (of Step 2)

Step 3: *If* $k > 0$, *go to Step 1. Otherwise, set*

$$J_{T_i^2} = \xi_{j,k}^T cov^{-1}\left(y_i - \hat{y}_i\right)\xi_{j,k}, \xi_{j,k} = y_i - \hat{y}_i;$$

Step 4: *Run detection logic*

$$\begin{cases} J_{T_i^2} - J_{th,i} \leq 0 \Longrightarrow fault - free, \\ J_{T_i^2} - J_{th,i} > 0 \Longrightarrow faulty, \end{cases}$$
$$J_{th,i} = \chi_\alpha^2 (m_i).$$

14.8 Combined Application of the Consensus and Correlation Based Schemes

Remember that in our previous study on average consensus based fault detection schemes no correlation between the process variables at different sub-processes has been assumed. Also, in our work on fault detection for dynamic processes, no process input variables and process noises have been taken into account. All these handlings may considerably limit practical applications of the proposed fault detection schemes. This motivates us to remove those assumptions and propose a modified scheme.

14.8.1 Models and Problem Formulation

Without loss of generality, we only consider dynamic processes and extend the models (14.54)–(14.55) to

$$x\left(k+1\right) = Ax(k) + Bu(k) + w(k), x(0) = x_0, \tag{14.104}$$
$$y(kT_{s,i}) = C_i x(kT_{s,i}) + v_i(kT_{s,i}) \in \mathcal{R}^{m_i}, i = 1, \cdots, M, \tag{14.105}$$

where $w(k) \sim \mathcal{N}(0, \Sigma_w)$ is the process noise vector and uncorrelated with v_i as well as $u(k), x(k)$,

$$\mathcal{E}v_{i,l}(\xi)v_{j,l}^{T}(\xi) = \mathcal{E}\begin{bmatrix} v_i\left(\xi T\right) \\ v_i\left(\xi T + \gamma_i T_s\right) \\ \vdots \\ v_i\left(\xi T + \left(\eta_i - 1\right)\gamma_i T_s\right) \end{bmatrix}\begin{bmatrix} v_j\left(\xi T\right) \\ v_j\left(\xi T + \gamma_j T_s\right) \\ \vdots \\ v_j\left(\xi T + \left(\eta_j - 1\right)\gamma_j T_s\right) \end{bmatrix}^T$$
$$= \begin{bmatrix} \Sigma_{v,ij} & & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \Sigma_{v,ij}\delta_{k_i,k_j} & \\ \cdots & & \ddots \end{bmatrix} =: \Sigma_{v_l,ij},$$
$$k_i \in \left\{\gamma_i, \cdots, \left(\eta_i - 1\right)\gamma_i\right\}, k_j \in \left\{\gamma_j, \cdots, \left(\eta_j - 1\right)\gamma_j\right\}.$$

$u(k)$ is the process input vector satisfying

$$x_u\left(k+1\right) = A_u x_u(k) \in \mathcal{R}^{n_u}, x_u(0) = v_{ref}, u(k) = C_u x_u(k) \tag{14.106}$$

with v_{ref} denoting the reference value that varies slowly.

Remark 14.10 *Dynamic system (14.106) can be viewed as a feed-forward controller. In case that a feedback control system is addressed, the state vector x and matrix A represent the closed-loop dynamics.*

Let

$$\bar{x}(k) = \begin{bmatrix} x(k) \\ x_u(k) \end{bmatrix}, \bar{A} = \begin{bmatrix} A & BC_u \\ 0 & A_u \end{bmatrix}, E = \begin{bmatrix} I \\ 0 \end{bmatrix}, \bar{C}_i = \begin{bmatrix} C_i & 0 \end{bmatrix}.$$

It turns out

$$\begin{aligned} \bar{x}\left(k+1\right) &= \bar{A}\bar{x}(k) + Ew(k), \\ y_i(kT_{s,i}) &= \bar{C}_i\bar{x}(kT_{s,i}) + v_i(kT_{s,i}), i = 1, \cdots, M. \end{aligned} \tag{14.107}$$

It is straightforward by the same lifting handling, as performed in Sect. 14.5, that the dynamics of the lifted system is governed by

$$\begin{aligned}
\bar{x}\left(\xi+1\right) =& \bar{A}_l \bar{x}(\xi) + E_l w_l(\xi), \bar{x}(\xi) = \bar{x}\left(\xi T\right), \bar{A}_l = \bar{A}^{\eta}, \qquad (14.108)\\
E_l =& \begin{bmatrix} \bar{A}^{\eta-1}E & \cdots & E \end{bmatrix}, w_l(\xi) = \begin{bmatrix} w(\xi T) \\ w(\xi T + T_s) \\ \vdots \\ w((\xi+1)\,T - T_s) \end{bmatrix},\\
y_{i,l}(\xi) =& \bar{C}_{i,l}\bar{x}(\xi) + F_{i,l}w_l(\xi) + v_{i,l}(\xi), \bar{C}_{i,l} = \begin{bmatrix} \bar{C}_i \\ \bar{C}_i\bar{A}^{\gamma_i} \\ \vdots \\ \bar{C}_i\bar{A}^{(\eta_i-1)\gamma_i} \end{bmatrix},\\
F_{i,l} =& \begin{bmatrix} 0 & & \cdots & & & & 0 \\ C_iA^{\gamma_i-1}E & \cdots & C_iAE & C_iE & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\ C_iA^{(\eta_i-1)\gamma_i-1}E & C_iA^{(\eta_i-1)\gamma_i-2}E & \cdots & \cdots & C_iAE & C_iE & 0 \end{bmatrix}
\end{aligned}$$

for $i = 1, \cdots, M$. The output model can be summarised as

$$y_l(\xi) = \bar{C}_l\bar{x}(\xi) + F_l w_l(\xi) + v_l(\xi), \bar{C}_l = \begin{bmatrix} \bar{C}_{1,l} \\ \vdots \\ \bar{C}_{M,l} \end{bmatrix}, F_l = \begin{bmatrix} F_{1,l} \\ \vdots \\ F_{M,l} \end{bmatrix}, \qquad (14.109)$$

which becomes, in the faulty case,

$$y_l(\xi) = \bar{C}_l\bar{x}(\xi) + H_f f_l(\xi) + F_l w_l(\xi) + v_l(\xi). \qquad (14.110)$$

Note that

$$\begin{aligned}
&\mathcal{E}\left(F_l w_l(\xi) + v_l(\xi)\right)\left(F_l w_l(\xi) + v_l(\xi)\right)^T \\
=& F_l \mathcal{E} w_l(\xi) w_l^T(\xi) F_l^T + \mathcal{E} v_l(\xi) v_l^T(\xi) =: \Sigma_{y_l},\\
&\mathcal{E} w_l(\xi) w_l^T(\xi) = diag\left(\Sigma_w, \cdots, \Sigma_w\right) =: \Sigma_{w_l},\\
&F_l \Sigma_{w_l} F_l^T = \begin{bmatrix} \ddots & & \vdots \\ & \Sigma_{w_l,ij} & \\ \vdots & & \ddots \end{bmatrix}, \Sigma_{w_l,ij} = F_{i,l}\Sigma_{w_l}F_{j,l}^T, i, j = 1, \cdots, M,\\
&\mathcal{E} v_l(\xi) v_l^T(\xi) = \begin{bmatrix} \ddots & & \vdots \\ & \Sigma_{v_l,ij} & \\ \vdots & & \ddots \end{bmatrix} =: \Sigma_{v_l},\\
&\mathcal{E}\left(w_l(\xi)\left(F_l w_l(\xi) + v_l(\xi)\right)^T\right) = \Sigma_{w_l}F_l^T.
\end{aligned}$$

14.8.2 Distributed Kalman Filter Based Fault Detection Scheme

We are now in the position to derive the distributed Kalman filter based fault detection algorithm. We first introduce the standard Kalman filter based residual generator and the test statistic for the lifted system model (14.108)–(14.109), which are, due to the correlation between the noises in the output and state models, different from the ones given in (14.63)–(14.67) and given as follows:

- Kalman filter based residual generator:

$$\hat{\bar{x}}\,(\xi+1)=\bar{A}_l\hat{\bar{x}}(\xi)+L_{kal}r_l(\xi), \tag{14.111}$$

$$r_l(\xi)=y_l(\xi)-\hat{y}_l(\xi),\ \hat{y}_l(\xi)=\bar{C}_l\hat{\bar{x}}(\xi),$$

$$L_{kal}=\left(\bar{A}_lP\bar{C}_l^T+E_l\Sigma_{w_l}F_l^T\right)\Sigma_r^{-1}, \tag{14.112}$$

$$\Sigma_r=\mathcal{E}\left(r_l(\xi)r_l^T(\xi)\right)=\bar{C}_lP\bar{C}_l^T+\Sigma_{y_l}$$

with P as the solution of Riccati equation

$$P=\bar{A}_lP\bar{A}_l^T+E_l\Sigma_{w_l}E_l^T-L_{kal}\Sigma_rL_{kal}^T;$$

- Test statistic and the corresponding threshold for detecting a fault occurring in the time interval $[\xi T-T_s,(\xi+1)T-T_s)$:

$$J=r_l^T(\xi)\left(H_f^T\Sigma_r^{-1}\right)^T\left(H_f^T\Sigma_r^{-1}H_f\right)^{-1}H_f^T\Sigma_r^{-1}r_l(\xi), \tag{14.113}$$

$$J_{th}=\chi_\alpha^2\left(k_f\right),k_f=\eta\left(n+n_u\right). \tag{14.114}$$

Next, we investigate a distributed realisation of (14.111)–(14.114). Again, the realisation is divided into two parts: distributed training (learning) and distributed online operation. It is evident that for our purpose consensus should be achieved for the matrices like

$$\bar{C}_l^T\Sigma_r^{-1},\bar{C}_l^T\Sigma_r^{-1}\bar{C}_l,F_l^T\Sigma_r^{-1},F_l^T\Sigma_r^{-1}F_l,H_f^T\Sigma_r^{-1},H_f^T\Sigma_r^{-1}H_f.$$

We first consider Σ_r^{-1}, $\Sigma_{y_l}^{-1}$, which can be re-written as

$$\Sigma_r^{-1}=\left(\bar{C}_lP\bar{C}_l^T+\Sigma_{y_l}\right)^{-1}=\Sigma_{y_l}^{-1}-\Sigma_{y_l}^{-1}\bar{C}_l\left(P^{-1}+\bar{C}_l^T\Sigma_{y_l}^{-1}\bar{C}_l\right)^{-1}\bar{C}_l^T\Sigma_{y_l}^{-1},$$
$$\Sigma_{y_l}^{-1}=\left(F_l\Sigma_{w_l}F_l^T+\Sigma_{v_l}\right)^{-1}=\Sigma_{v_l}^{-1}-\Sigma_{v_l}^{-1}F_l\left(\Sigma_{w_l}^{-1}+F_l^T\Sigma_{v_l}^{-1}F_l\right)^{-1}F_l^T\Sigma_{v_l}^{-1}.$$

It yields

$$\bar{A}_l P \bar{C}_l^T \Sigma_r^{-1} = \bar{A}_l \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1}, \tag{14.115}$$

$$E_l \Sigma_{w_l} F_l^T \Sigma_r^{-1} = E_l \Sigma_{w_l} F_l^T \left(\Sigma_{y_l}^{-1} - \Sigma_{y_l}^{-1} \bar{C}_l \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1}\right), \tag{14.116}$$

$$H_f^T \Sigma_r^{-1} = H_f^T \Sigma_{y_l}^{-1} - H_f^T \Sigma_{y_l}^{-1} \bar{C}_l \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1}, \tag{14.117}$$

$$F_l^T \Sigma_{y_l}^{-1} = \Sigma_{w_l}^{-1} \left(\Sigma_{w_l}^{-1} + F_l^T \Sigma_{v_l}^{-1} F_l\right)^{-1} F_l^T \Sigma_{v_l}^{-1}, \tag{14.118}$$

$$\bar{C}_l^T \Sigma_{y_l}^{-1} = \bar{C}_l^T \Sigma_{v_l}^{-1} - \bar{C}_l^T \Sigma_{v_l}^{-1} F_l \left(\Sigma_{w_l}^{-1} + F_l^T \Sigma_{v_l}^{-1} F_l\right)^{-1} F_l^T \Sigma_{v_l}^{-1},$$

$$H_f^T \Sigma_{y_l}^{-1} = H_f^T \Sigma_{v_l}^{-1} - H_f^T \Sigma_{v_l}^{-1} F_l \left(\Sigma_{w_l}^{-1} + F_l^T \Sigma_{v_l}^{-1} F_l\right)^{-1} F_l^T \Sigma_{v_l}^{-1}. \tag{14.119}$$

We now focus on the computation of $\bar{C}_l^T \Sigma_{v_l}^{-1}$, $F_l^T \Sigma_{v_l}^{-1}$, $H_f^T \Sigma_{v_l}^{-1}$, which builds the core of the above matrix computations. Let

$$\bar{C}_l^T \Sigma_{v_l}^{-1} = \left[\, \varGamma_{C,1} \cdots \varGamma_{C,M} \,\right] =: \varGamma_C,\ F_l^T \Sigma_{v_l}^{-1} = \left[\, \varGamma_{F,1} \cdots \varGamma_{F,M} \,\right] =: \varGamma_F,$$
$$H_f^T \Sigma_{v_l}^{-1} = \left[\, \varGamma_{H,1} \cdots \varGamma_{H,M} \,\right] =: \varGamma_H.$$

It turns out

$$\begin{bmatrix} \varGamma_{C,1} & \cdots & \varGamma_{C,M} \\ \varGamma_{F,1} & \cdots & \varGamma_{F,M} \\ \varGamma_{H,1} & \cdots & \varGamma_{H,M} \end{bmatrix} \Sigma_{v_l} = \begin{bmatrix} \bar{C}_{1,l}^T & \cdots & \bar{C}_{M,l}^T \\ F_{1,l}^T & \cdots & F_{M,l}^T \\ H_{1,f}^T & \cdots & H_{M,f}^T \end{bmatrix}. \tag{14.120}$$

Since Σ_{v_l} is, due to the correlations between the nodes, not diagonal, we propose, on the assumption that those correlated nodes are also connected, to apply the distributed iteration learning algorithm given in Sub-section 14.6.2 to determine $\varGamma_C, \varGamma_F, \varGamma_H$. We would like to call reader's attention that the necessary data for running iteration learning algorithm are the local data. That is, at the i-th node only $\bar{C}_{i,l}^T$, $F_{i,l}^T$, $H_{i,f}^T$ as well as the i-th row block of Σ_{v_l} are needed. As a result (the output of the algorithm), $\varGamma_{C,i}, \varGamma_{F,i}, \varGamma_{H,i}$ are available at node i, $i = 1, \cdots, M$.

Next, applying the average consensus algorithm delivers

$$\frac{1}{M}\sum_{i=1}^{M} \varGamma_{C,i} \bar{C}_{i,l} = \frac{1}{M} \bar{C}_l^T \Sigma_{v_l}^{-1} \bar{C}_l,\ \frac{1}{M}\sum_{i=1}^{M} \varGamma_{F,i} F_{i,l} = \frac{1}{M} F_l^T \Sigma_{v_l}^{-1} F_l,$$
$$\frac{1}{M}\sum_{i=1}^{M} \varGamma_{H,i} H_{i,f} = \frac{1}{M} H_f^T \Sigma_{v_l}^{-1} H_f,\ \frac{1}{M}\sum_{i=1}^{M} \varGamma_{F,i} \bar{C}_{i,l} = \frac{1}{M} F_l^T \Sigma_{v_l}^{-1} \bar{C}_l,$$
$$\frac{1}{M}\sum_{i=1}^{M} \varGamma_{H,i} \bar{C}_{i,l} = \frac{1}{M} H_f^T \Sigma_{v_l}^{-1} \bar{C}_l,\ \frac{1}{M}\sum_{i=1}^{M} \varGamma_{F,i} H_{i,f} = \frac{1}{M} F_l^T \Sigma_{v_l}^{-1} H_f,$$

at all nodes, which allow all nodes to compute $\bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l$, $F_l^T \Sigma_{y_l}^{-1} F_l$, $\bar{C}_l^T \Sigma_{y_l}^{-1} F_l$ and further to solve Riccati equation,

$$\begin{aligned}P = \bar{A}_l P \bar{A}_l^T + E_l \Sigma_{w_l} E_l^T - L_{kal} \Sigma_r L_{kal}^T \Longrightarrow \quad (14.121)\\
L_{kal} \Sigma_r L_{kal}^T = \left(\bar{A}_l P \bar{C}_l^T + E_l \Sigma_{w_l} F_l^T\right) \Sigma_r^{-1} \left(\bar{A}_l P \bar{C}_l^T + E_l \Sigma_{w_l} F_l^T\right)^T,\\
\bar{C}_l^T \Sigma_r^{-1} \bar{C}_l = P^{-1} \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l,\\
F_l^T \Sigma_r^{-1} F_l = F_l^T \Sigma_{y_l}^{-1} F_l - F_l^T \Sigma_{y_l}^{-1} \bar{C}_l \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1} F_l,\\
F_l^T \Sigma_r^{-1} \bar{C}_l = F_l^T \Sigma_{y_l}^{-1} \bar{C}_l - F_l^T \Sigma_{y_l}^{-1} \bar{C}_l \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\end{aligned}$$

for P at all nodes. Let

$$L_{kal} = \left[\, L_{kal,1} \cdots L_{kal,M} \,\right], H_r = H_f^T \Sigma_r^{-1} = \left[\, H_{r,1} \cdots H_{r,M} \,\right].$$

The final step in the training process is to calculate $L_{kal,i}$, $H_{r,i}$, respectively,

$$\begin{aligned}L_{kal} = \bar{A}_l P \bar{C}_l^T \Sigma_r^{-1} + E_l \Sigma_{w_l} F_l^T \Sigma_r^{-1} =\\
\left(\bar{A}_l - E_l \Sigma_{w_l} F_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right) \Pi \bar{C}_l^T \Sigma_{y_l}^{-1} + E_l \Sigma_{w_l} F_l^T \Sigma_{y_l}^{-1} \Longrightarrow\\
L_{kal,i} = \left(\bar{A}_l - E_l \Sigma_{w_l} F_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right) \Pi \left(\Gamma_{C,i} - \Pi_{C,i} \Gamma_{F,i}\right) + \Pi_{E,i} \Gamma_{F,i}, \quad (14.122)\\
\Pi = \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1}, \Pi_{E,i} = E_l \left(\Sigma_{w_l}^{-1} + F_l^T \Sigma_{v_l}^{-1} F_l\right)^{-1} F_l^T \Sigma_{v_l}^{-1},\\
\Pi_{C,i} = \bar{C}_l^T \Sigma_{v_l}^{-1} F_l \left(\Sigma_{w_l}^{-1} + F_l^T \Sigma_{v_l}^{-1} F_l\right)^{-1},\\
H_r = H_f^T \Sigma_r^{-1} = H_f^T \Sigma_{y_l}^{-1} - H_f^T \Sigma_{y_l}^{-1} \bar{C}_l \Pi \bar{C}_l^T \Sigma_{y_l}^{-1} \Longrightarrow\\
H_{r,i} = \Gamma_{H,i} - \Pi_{H,i} \Gamma_{F,i} - H_f^T \Sigma_{y_l}^{-1} \bar{C}_l \Pi \left(\Gamma_{H,i} - \Pi_{C,i} \Gamma_{F,i}\right), \quad (14.123)\\
\Pi_{H,i} = H_f^T \Sigma_{v_l}^{-1} F_l \left(\Sigma_{w_l}^{-1} + F_l^T \Sigma_{v_l}^{-1} F_l\right)^{-1}\end{aligned}$$

at node i, $i = 1, \cdots, M$, and

$$H_f^T \Sigma_r^{-1} H_f = H_f^T \Sigma_{y_l}^{-1} H_f - H_f^T \Sigma_{y_l}^{-1} \bar{C}_l \left(P^{-1} + \bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l\right)^{-1} \bar{C}_l^T \Sigma_{y_l}^{-1} H_f \tag{14.124}$$

at all nodes using the available $H_f^T \Sigma_{y_l}^{-1} H_f$, $H_f^T \Sigma_{y_l}^{-1} \bar{C}_l$, $\bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l$, P. For the online implementation, the average consensus algorithm is applied to perform

$$L_{kal} r_l(\xi) = \sum_{i=1}^{M} L_{kal,i} r_{i,l}(\xi), H_f^T \Sigma_r^{-1} r_l(\xi) = \sum_{i=1}^{M} H_{r,i} r_{i,l}(\xi),$$

and further the Kalman filter based residual generator (14.111) as well as the test statistic (14.113).

14.8.3 Training and Online Implementation Algorithms

We summarise the above results in the following two algorithms.

Algorithm 14.12 *The training algorithm for distributed Kalman filter based fault detection with correlated measurement noises*

Step 0: Identify the covariance matrix of measurement noises Σ_{v_l} running parallel at nodes $i = 1, \cdots, M$. At node i, the corresponding i-th row block of Σ_{v_l} is available;

Step 1: Run the distributed iteration learning algorithm given in Sub-section 14.6.2 to solve (14.120) for

$$\Gamma_C = \bar{C}_l^T \Sigma_{v_l}^{-1}, \Gamma_H = F_l^T \Sigma_{v_l}^{-1}, \Gamma_H = H_f^T \Sigma_{v_l}^{-1};$$

Step 2: Run the average consensus algorithm to build

$$\bar{C}_l^T \Sigma_{v_l}^{-1} \bar{C}_l, F_l^T \Sigma_{v_l}^{-1} F_l, H_f^T \Sigma_{v_l}^{-1} H_f, F_l^T \Sigma_{v_l}^{-1} \bar{C}_l, H_f^T \Sigma_{v_l}^{-1} \bar{C}_l, F_l^T \Sigma_{v_l}^{-1} H_f$$

at all nodes;

Step 3 Calculate $\bar{C}_l^T \Sigma_{y_l}^{-1} \bar{C}_l, F_l^T \Sigma_{y_l}^{-1} F_l, \bar{C}_l^T \Sigma_{y_l}^{-1} F_l$ at all nodes, according to (14.118)-(14.119);

Step 4 Solve Riccati equation(14.121)for P at all nodes;

Step 5: Calculate $L_{kal,i}, H_{r,i}$ according to (14.122) and (14.123) in parallel at nodesi $= 1, \cdots, M$, and $H_f^T \Sigma_r^{-1} H_f$ according to (14.124) and output them.

Algorithm 14.13 *The online implementation algorithm for distributed Kal-man filter based fault detection with correlated measurement noises*

Step 0: Compute

$$r_{i,KF}(\xi) = L_{kal,i} r_{i,l}(\xi), r_{j,J}(\xi) = H_{r,i} r_{i,l}(\xi)$$

in parallel at nodes $i = 1, \cdots, M$;

Step 1: Start an average consensus algorithm to compute

$$\bar{r}_{KF}(\xi) = \frac{1}{M} \sum_{j=1}^{M} r_{j,KF}(\xi), \bar{r}_J(\xi) = \frac{1}{M} \sum_{i=1}^{M} r_{j,J}(\xi);$$

Step 2: Calculate

$$\hat{x}_i(\xi + 1) = \bar{A}_l \hat{x}_i(\xi) + \bar{r}_{KF}(\xi), J_i = M \bar{r}_J^T(\xi) \left(H_f^T \Sigma_r^{-1} H_f \right)^{-1} \bar{r}_J(\xi)$$

in parallel at nodes $i = 1, \cdots, M$;

Step 3: *Check*

$$J_i - J_{th} = J_i - \chi^2_\alpha\left(k_f\right)$$

for decision

$$J_i - J_{th} \leq 0 \Longrightarrow \textit{fault-free, otherwise faulty and alarm}$$

at all nodes $i = 1, \cdots, M$;

Step 4: *Output*

$$\hat{x}_i\left(\xi + 1\right), \textit{and in faulty case alarm}$$

at nodes $j = 1, \cdots, M$.

14.9 Notes and References

Fault detection in large-scale, interconnected and distributed systems is a challenging issue that will certainly become one of the dominant topics in the fault diagnosis research and application areas in this and the next decade. Our study in this chapter has been dedicated to two different classes of large-scale processes: (i) large-scale processes equipped with a distributed sensor (monitoring) network and (ii) interconnected large-scale processes with weakly coupled sub-processes. Correspondingly, the fault detection objectives are different as well. For the first class of processes, the focus is on detecting the faults within the process (as a whole) from various sensor nodes (locations) which are distributed and networked. The basic idea behind that is to increase fault detection performance by means of redundant sensors and fusion of sensor data. Differently, the objective of fault detection in the second class of processes consists in detecting faults in each sub-processes. To this end, the local fault detection systems located at the sub-processes are networked corresponding to the coupling/correlation topology, in order to exchange information among the local fault detection systems. This allows an optimised utilisation of system correlations towards an optimal fault detection.

Although our main attention in this chapter has been paid to the issues of distributed fault detection, the formulated fault detection problems have been addressed in the data-driven and statistic framework. In this context, the handled fault detection problems are generally solved in two steps: (i) data collection and pre-processing, which is often understood as training or learning and thus performed offline, and (ii) online fault detection. Reviewing the existing publications on the topic of distributed fault detection in networked systems and under consideration of network topology shows evidently that the major focus in this research domain is on the design of distributed fault detection systems towards distributed online fault detection. Consequently, model-based methods are mainly applied. So far, our work in this chapter is different and dedicated both to offline learning and online fault detection in a distributed fashion. At this point, it should be mentioned that a number of data-driven fault detection approaches have been reported under the heading of distributed methods, although they have mainly addressed distributed or parallel computations, and not taken into account the data transmissions among the sub-processes or sensor nodes and their influences on fault detection performance. The major focus of these methods is mainly on processing of "big data". From the viewpoint of gathering information, these methods often lead to a centralised fault detection. This is the major difference to our work.

It is the state of the art that network topology plays a considerable role in today's research investigations on distributed process monitoring, diagnosis and control. For this reason, at the beginning of this chapter, basic definitions and concepts in network and graph theory have been introduced. The reader is referred to, for instance, [1, 2] for a systematic and detailed introduction.

For detecting faults in a large-scale process by means of a distributed sensor network, we have adopted the average consensus technique for a fusion of process data received by the sensor network. The average consensus technique has been successfully applied to data fusion, state estimation in distributed systems, multi-agent systems, and become a well-established tool to deal with distributed system issues [3]. The introduction to the basics of the average consensus technique in Subsection 14.1.2 with Theorem 14.1 as the main result is given in the highly cited paper on distributed average consensus by Xiao and Boyd [4]. The construction of the weighting matrix W given in (14.11) and (14.14) can be found in [4] and [5], respectively. For further methods, the reader is referred to the survey paper [3].

Applying the average consensus algorithm, we have developed two basic fault detection schemes as well as a number of their variations for large-scale processes equipped with distributed sensor networks. The first one is an intuitive realisation of average consensus based the fault detection idea, in which the average consensus algorithm leads to the availability of the average of all sensor measurements at each sensor node. Considering that the average of all sensor variables suppresses uncertainty and variation in the measurements at average, it is expected that the fault detectability can be enhanced. It is remarkable that the implementation of this fault detection scheme requires considerably reduced data transmissions in comparison with other consensus-based methods. In particular, during the training phase, only local computations are to be performed and no data transmission is necessary. On the other hand, this fault detection scheme does not result in optimal performance in the sense of optimal fault detection formulated in Sect. 3.2. Alternatively, we have proposed the second fault detection scheme that gives a "distributed version" of the optimal solution to the detection problem defined in Sect. 3.2. This fault detection scheme consists of two algorithms: a distributed offline learning/training and a distributed online fault detection. For both of these algorithms, the average consensus builds the core computation. In comparison to the first scheme, the data transmissions between the sensor nodes increase significantly.

Although the application of the consensus algorithm can significantly improve the fault detectability by an optimal data fusion, the iteration computations to be performed to reach consensus cause delays in fault detection. To deal with this problem, different variations of the above-mentioned fault detection schemes have been proposed. Among them, the idea of performing fault detection at each iteration during the consensus computation has been, to our best knowledge, proposed for the first time and realised in the consensus manner as well. The core of this idea is the computation or estimation of the covariance matrix of the random vector being available at each sensor node during the consensus iteration. In this way, T^2-test statistic and further detection logic can be implemented at each node during the iteration. As a result, delays caused by consensus iterations can be significantly reduced. It is remarkable that in this context the intuitive fault detection scheme is more efficient. It requires less online data transmission and computations on the one hand, and allows an exact computation of variance of the test statistic on the other hand, which results in obviously an optimal fault detection.

On the assumption of an available state space process model, which can be identified during the training phase using collected process data and by means of, for instance, data-driven SKR realisation schemes introduced in Chap. 4, our consensus based fault detection schemes have been extended to detecting faults in large-scale dynamic processes equipped with sensor networks. The core of this extension is the application of a distributed Kalman filter. Consensus based distributed Kalman filters are a well-established technique and widely applied to data fusion and distributed estimation. In an early and highly cited conference paper, Olfati-Saber has proposed the structure and a design scheme of distributed Kalman filters [6], in which average consensus is performed for the fusion of measurement data/innovations generated at all sensor nodes and the update (iteration) of covariance matrix of state estimation error vector. There are a series of follow-up publications which have adopted the essential structure of the distributed Kalman filter proposed in [6] and extended the algorithm to different variations [3, 7–9]. It is natural that such a distributed Kalman filter is applied for residual generation and, based on it, further for fault detection. On the other hand, in order to achieve an optimal fault detection, a special fusion of the local residual vectors becomes necessary, as formulated in Sect. 3.2. Under consideration of our objective, we have proposed a consensus Kalman filter based distributed fault detection scheme, in which the structure of the distributed Kalman filter proposed in [6] has been adopted, and in addition, an optimal fusion of the residual vectors from the sensor nodes is implemented. From the technical implementation point of view, we have introduced a lifting model for the dynamic process under consideration, based on which the proposed distributed Kalman filter is realised. Consequently, this allows a reliable and timely well-synchronised implementation of the consensus-based fault detection, as schematically sketched in Fig. 14.2. Moreover, the determination of the covariance matrices of the state estimation errors and the residual vector is performed distributed using average consensus algorithm and during the offline learning phase, so that the online computations can be reduced.

The objective of our study on fault detection in interconnected large-scale processes with weakly coupled sub-processes is to improve the fault detectability by making use of correlation relations among the sub-processes. Different from our work in the first part of this chapter, the focus in this part of work is on detecting faults in sub-processes. In this context, the tasks can be formulated as those optimal fault detection problems formulated in Sect. 3.2 and Sub-section 3.3.3. As revealed in Theorem 14.5, the optimal solution can be equivalently expressed as a least mean squares (LMS) estimation of the measurement vector, for instance, the i-th node by means of the correlated measurement vectors from other nodes. The remaining works in this part have been dedicated to the realisation of an LMS estimate in a distributed interconnected large-scale process, whose communication topology is coincident with the correlation structure of the process under consideration.

Distributed LS (least squares) or LMS estimation issues are a thematic field that receives considerable research interests both in communication and control communities [10, 11]. There are numerous strategies to achieve distributed estimation. For instance, the distributed estimation computations are coordinated by a fusion center [12] or the consensus strategy, as introduced at the beginning of this chapter, can be adopted. We have decided to follow the strategy that the LMS estimate is performed iteratively, distributed and the involved nodes will only deliver a part or a sub-set of the estimate. In other words, it is not our intention that all nodes should share an identical estimate or process knowledge, as the consensus strategy does. This decision is motivated by the facts that

- different sub-processes could have significantly different correlation structures and thus there is no need for each node to share identical process knowledge,
- the large-scale of such interconnected processes, other than their counterpart addressed in the first part of our study, demands for distributed computations to achieve collective process knowledge, and
- in the context of data-driven fault detection, the computationally intensive and involved solution of the LMS problem is performed in the training phase. In this sense, the real-time requirement on involved computations is low.

Inspired by the idea in [11] to re-formulate an LMS estimate problem as the solution of linear equation which is then solved iteratively, we have formulated our task of finding the distributed regression model (14.78)–(14.79) for the LMS estimation as solving a group of linear equations distributed and iteratively, as formulated by (14.80). In fact, the distributed solution of (14.80) is the Richardson's method that is well-established in the theoretical framework of parallel computations [13]. The results given in Proposition 14.1 and condition (14.84) can be found in [13].

Aiming at reducing online communication and computation for an LMS estimate of the measurement vector of a sub-process (a node), further investigation has been done on the relation between the distance and the intensity of the correlations between the nodes. It has been demonstrated by (14.95) that with the increasing distance between two nodes the correlation between these two nodes will become weaker. In other words, those nodes, which are far away (in the sense of distance) from a node, say the i-th node, will contribute less to the estimate of the measurement vector of the i-th sub-process. This motivates us to propose two alternative schemes for the LMS estimation. Both of them are approximated solutions of the optimal estimate using the data from those nodes, which are located close (in the sense of the node distance) to the node whose measurement vector is to be estimated. In this way, both the computation and communication load can be (significantly) reduced.

At the end of this chapter, we have re-studied the average consensus based fault detection issues, in which the correlations between the sub-processes/nodes are now taken into account. Although we have only addressed the fault detection issues for dynamic processes using distributed Kalman filters, the application to static processes is straightforward. The idea behind this work is to realise the needed computation of the covariance matrix of the measurement noises (of the overall process), which is, due to the correlations among the sub-processes, no more diagonal matrix and thus becomes computationally involved, using the iteration algorithm adopted for the LMS estimation. Because the iterative computation of this algorithm is performed during the training phase, the expected online communication and computation loads for performing the fault detection algorithm, including the consensus based distributed Kalman filter and test statistics, are similar to the ones in case of no correlation being under consideration.

We would like to emphasise that most of the methods and algorithms proposed in our work can be applied to dealing with distributed fault detection in processes with (deterministic) unknown inputs and disturbances. In other words, the fault detection problems formulated in Sect. 2.3 for static processes and the unified solution for dynamic processes presented in Sect. 4.3 can be realised in a distributed fashion, as we studied in this chapter for processes with noises. The needed modifications and extensions are straightforward.

References

1. R. Diestel, *Graph Theory*. Berlin, New York: Springer-Verlag, 2005.
2. C. Godsil and G. Royle, *Algebraic Graph Theory, Graduate Texts in Mathematics*. relax Berlin: Springer, 2001.
3. R. Olfati-Saber, A. Fax, and R. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, pp. 215–233, 2007.
4. L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.
5. L. Xiao, S. Boyd, and S.-J. Kim, “Distributed average consensus with least-mean-square deviation,” *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, 2007.
6. R. Olfati-Saber, “Distributed kalman filter with embedded consensus filters,” *Proc. of the 44th IEEE CDC*, pp. 8179–8184, 2005.
7. R. Olfati-Saber, “Distributed kalman filtering for sensor networks,” *Proc. of the 46th IEEE CDC*, pp. 5492–5498, 2007.
8. E. Song, Y. Zhu, J. Zhou, and Z. You, “Optimal kalman filtering fusion with cross-correlated sensor noises,” *Automatica*, vol. 43, pp. 1450–1456, 2007.
9. I. Matei and J. Baras, “Consensus-based linear distributed filtering,” *Automatica*, vol. 48, pp. 1776–1782, 2012.
10. F. Cattivelli and A. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Trans. on Signal Processing*, vol. 58, pp. 1035–1048, 2010.
11. D. E. Marelli and M. Fu, “Distributed weighted least-squares estimation with fast convergence for large-scale systems,” *Automatica*, vol. 51, pp. 27–39, 2015.
12. J. Fang and H. Li, “Joint dimension assignment and compression for distributed multisensor estimation,” *IEEE Signal Processing Letters*, vol. 15, pp. 174–177, 2008.
13. D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

Chapter 15
Alternative Test Statistics and Fault Detection Schemes

In Sect. 2.2, we have formulated optimal fault detection problems in the statistic framework and derived numerous solutions in the subsequent works. Roughly speaking, most of these solutions could deliver optimal fault detection performance when the faults under consideration only cause changes in the mean value (vector) of the measurement variables. It can be noticed that the optimal performance is achieved by the use of χ^2- or T^2-test statistics. In fact, these two test statistics are mostly applied ones in fault diagnosis research and practice. Because they are so popular and viewed as well-established, only very few users care about (i) the idea behind them, (ii) on which assumptions they could be applied successfully, and (iii) what is the achievable performance. In Chap. 3, we have discussed about these issues in detail and demonstrated that

- χ^2- or T^2-test statistics are the result of applying the generalised likelihood ratio method to detecting changes in mean caused by the faults,
- on the assumption that the process measurement (vector) is normally distributed with constant mean and covariance, and
- the fault detection performance is optimal in the sense that at an acceptable level of false alarm rate the fault detection rate (probability) reaches maximum.

These facts raise, on the other hand, questions concerning the fault detection performance and the test statistics used for decision making, when the assumptions are not satisfied, for instance, due to distribution other than normal distribution or varying mean or changes in the covariance caused by the faults. A further concern is the result of the observation that a metric measuring the distance between two probabilistic distributions is often adopted to build a test statistic for decision making. Should we follow Neyman-Pearson Lemma and apply GLR methods for the determination of the test statistic or apply a metric as the test statistic? Which one of these two strategies is the optimal one? These questions are the background and motivations as well for our subsequent investigations.

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_15

15.1 A General Formulation and Solution of GLR-based Fault Detection

Consider measurement vector $y \in \mathcal{R}^m$, which is a random vector with (known) probability density function (PDF) $f_\theta(y)$. Here, θ denotes the parameter set of the PDF. Suppose that the faults under consideration cause changes in the PDF parameters and are modelled by

$$\theta = \begin{cases} \theta_0, & \text{fault-free (nominal)}, \\ \theta_f, & \text{faulty}. \end{cases} \tag{15.1}$$

Let

$$L\left(\theta \left| y \right.\right) = f_\theta(y) \tag{15.2}$$

be the likelihood function of θ given y. The likelihood ratio adopted in the GLR-based fault detection scheme is defined by

$$s\left(y\right) = \frac{L\left(\theta_f \left| y \right.\right)}{L\left(\theta_0 \left| y \right.\right)}. \tag{15.3}$$

Since θ_f is often unknown, θ_f in the LR $s(y)$ is substituted by its maximum likelihood estimate (MLE) $\hat{\theta}_f$. That is

$$\hat{\theta}_f = \arg\max_{\theta_f} L\left(\theta_f \left| y \right.\right) \Longleftrightarrow \hat{\theta}_f = \arg\max_{\theta_f} s\left(y\right). \tag{15.4}$$

It yields

$$s\left(y\right) = \frac{L\left(\hat{\theta}_f \left| y \right.\right)}{L\left(\theta_0 \left| y \right.\right)} =: J\left(y\right). \tag{15.5}$$

In the framework of fault detection, $J\left(y\right)$ is called test statistic. Note that $J\left(y\right)$ is a random variable and a function of y. Hence, for known $f_\theta(y)$ and some given number a, the probability

$$\Pr\left(J\left(y\right) \leq a\right)$$

can be, under certain conditions, calculated analytically or using some numerical methods, when $\hat{\theta}_f$ is available.

In the practice of fault detection, log-likelihood ratio is commonly adopted with n samples $y(i), i = 1, \cdots, n$. It yields

$$J = s_n(y) = \log \prod_{i=1}^{n} \frac{L\left(\hat{\theta}_f \left| y(i)\right.\right)}{L\left(\theta_0 \left| y(i)\right.\right)} = \sum_{i=1}^{n} \left(L_{\log}\left(\hat{\theta}_f \left| y(i)\right.\right) - L_{\log}\left(\theta_0 \left| y(i)\right.\right)\right), \tag{15.6}$$

$$L_{\log}\left(\hat{\theta}_f \left| y(i)\right.\right) = \log L\left(\hat{\theta}_f \left| y(i)\right.\right), L_{\log}\left(\theta_0 \left| y(i)\right.\right) = \log L\left(\theta_0 \left| y(i)\right.\right).$$

Next, we consider an example of the above GLR-based fault detection scheme.

Example 15.1 *Suppose*

$$y \in \mathcal{R}^m, y \sim \mathcal{N}\left(\mu, \Sigma_y\right)$$

with the PDF

$$f_\theta(y) = \frac{1}{\sqrt{(2\pi)^m \det\left(\Sigma_y\right)}} e^{-\frac{1}{2}(y-\mu)^T \Sigma_y^{-1}(y-\mu)},$$

$$\mu = \mathcal{E} y = \begin{cases} \mu_0, \textit{ fault-free (nominal)}, \\ \mu_f, \textit{ faulty}, \end{cases} \tag{15.7}$$

$$\Sigma_y = \begin{cases} \Sigma_0, \textit{ fault-free (nominal)}, \\ \Sigma_f, \textit{ faulty}. \end{cases} \tag{15.8}$$

Hence, the log-likelihood ratio is given by

$$\begin{aligned} s(y) &= \ln \frac{L\left(\mu_{f,} \Sigma_f \left| y\right.\right)}{L\left(\mu_{0,} \Sigma_0 \left| y\right.\right)} \\ &= \frac{1}{2}\left(\ln \frac{\det(\Sigma_0)}{\det\left(\Sigma_f\right)} + (y-\mu_0)^T \Sigma_0^{-1} (y-\mu_0) - \left(y-\mu_f\right)^T \Sigma_f^{-1} \left(y-\mu_f\right)\right) \end{aligned}$$

or in more general case with n measurement data $y(i), i = 1, \cdots, n$,

$$\begin{aligned} s_n(y) &= \ln \prod_{i=1}^{n} \frac{L\left(\mu_{f,} \Sigma_f \left| y(i)\right.\right)}{L\left(\mu_{0,} \Sigma_0 \left| y(i)\right.\right)} \\ &= \frac{1}{2}\left(n \ln \frac{\det(\Sigma_0)}{\det\left(\Sigma_f\right)} + \sum_{i=1}^{n} \left(\begin{array}{c} (y(i)-\mu_0)^T \Sigma_0^{-1} (y(i)-\mu_0) \\ -\left(y(i)-\mu_f\right)^T \Sigma_f^{-1} \left(y(i)-\mu_f\right) \end{array} \right)\right). \end{aligned} \tag{15.9}$$

It is well-known that the MLE estimates of μ_f, Σ_f *are*

$$\hat{\mu}_f = \frac{1}{n} \sum_{i=1}^{n} y(i), \hat{\Sigma}_f = \frac{1}{n} \sum_{i=1}^{n} \left(y(i) - \hat{\mu}_f\right)\left(y(i) - \hat{\mu}_f\right)^T,$$

respectively. As a result, the test statistic is defined as

$$J = n \ln \frac{\det\left(\Sigma_0\right)}{\det\left(\hat{\Sigma}_f\right)} + \sum_{i=1}^{n} \left(\begin{array}{c} \left(y\left(i\right) - \mu_0\right)^T \Sigma_0^{-1} \left(y\left(i\right) - \mu_0\right) \\ - \left(y\left(i\right) - \hat{\mu}_f\right)^T \hat{\Sigma}_f^{-1} \left(y\left(i\right) - \hat{\mu}_f\right) \end{array} \right). \tag{15.10}$$

Here, factor $\frac{1}{2}$ is omitted without causing changes in the fault detection performance. On the assumption that only changes in μ are considered, the test statistic becomes

$$J = \hat{\mu}_f^T \Sigma_0^{-1} \hat{\mu}_f = \left(\frac{1}{n} \sum_{i=1}^{n} y\left(i\right)\right)^T n \Sigma_0^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} y\left(i\right)\right), \tag{15.11}$$

which is the commonly used χ^2- or T^2-test statistic.

Remark 15.1 *In the data-driven fault detection framework, μ_0, Σ_0 are identified using sufficient number of process data. Also, by centering the process data, μ_0 can be assumed to be zero.*

In general, there is no analytical solution for the probability computation based on the LR-based test statistic $s_n\left(y\right)$ given in (15.6). As a consequence, threshold setting is a hard task. Alternatively, numerical solutions can be used. Below is an algorithm running offline (in the training phase) for the threshold setting.

Algorithm 15.1 *Threshold setting, when GLR is used as test statistic*

```
Step 0:   Set $J_{th} = 0$;
Step 1:   For $j = 1$ to $N$
    Generate n data from the underlying distribution with the PDF $f_{\theta_0}(y)$, $y\left(i\right), i = 1, \cdots, n$, using the so-called randomised algorithm;
    Calculate $\hat{\theta}_f$ and $J = s_n\left(y\right)$ according to (15.4) and(15.6), respectively;
    If
        $J > J_{th}$
    then set
        $J_{th} = J$;
    End.
Step 2:   Output $J_{th}$.
```

Remark 15.2 *In the next chapter, we shall introduce the randomised algorithm (RA) technique and its application to fault detection in more details.*

Recall that

$$J^{1/2} = \sqrt{\hat{\mu}_f^T \Sigma_0^{-1} \hat{\mu}_f}$$

in (15.11) is the so-called Mahalanobis distance which is a dissimilarity measure between two random vectors of the same distribution with the (same) covariance matrix Σ_0. In this context, the GLR-induced test statistic J can be interpreted as the dissimilarity measure between the mean value of the fault-free operation and the average value of the (present) real-time operation. A larger J is understood as higher dissimilarity. When J is larger than the threshold, the high dissimilarity is interpreted as the result of a fault. This observation inspires the idea of applying a dissimilarity measure between two distributions to a successful fault detection.

15.2 KL Divergence Based Fault Detection Schemes

We consider again the measurement vector $y \in \mathcal{R}^m$ with PDFs $f_{\theta_0}(y)$ and $f_{\theta_f}(y)$ representing the distributions in the fault-free and faulty operations, respectively. In statistics, Kullback-Leibler (KL) divergence is a well-established dissimilarity measure between two distributions and thus its application to fault detection has received considerable attention. In this section, we investigate KL divergence based fault detection issues.

15.2.1 On KL Divergence

KL divergence from the distribution denoted by f_{θ_f} with PDF $f_{\theta_f}(y)$ to the one f_{θ_0} with PDF $f_{\theta_0}(y)$ is defined and denoted by

$$D\left(f_{\theta_f}, f_{\theta_0}\right) = \int f_{\theta_f}(y) \log \frac{f_{\theta_f}(y)}{f_{\theta_0}(y)} dy. \tag{15.12}$$

The KL divergence has fundamental properties:

$$D\left(f_{\theta_f}, f_{\theta_0}\right) \geq 0, D\left(f_{\theta_f}, f_{\theta_0}\right) = 0, \text{ when } f_{\theta_f}(y) = f_{\theta_0}(y); \tag{15.13}$$

$$D\left(f_{\theta_f}, f_{\theta_0}\right) \neq D\left(f_{\theta_0}, f_{\theta_f}\right) = \int f_{\theta_0}(y) \log \frac{f_{\theta_0}(y)}{f_{\theta_f}(y)} dy. \tag{15.14}$$

Inequality (15.14) tells us, KL divergence is asymmetric and thus is not a metric. In fact, the asymmetry plays an important role in dealing with fault detection issues, which has, unfortunately, not received reasonable attention by reviewing the published results. This motivates us to address this issue in the sequel.

Recall that

$$\log \frac{f_{\theta_f}(y)}{f_{\theta_0}(y)} = \log \frac{L\left(\theta_f \left| y\right.\right)}{L\left(\theta_0 \left| y\right.\right)} = s\left(y\right)$$

is the log-likelihood ratio. Hence, the KL divergence from f_{θ_f} to f_{θ_0} can also be written as

$$D\left(f_{\theta_f}, f_{\theta_0}\right) = \int f_{\theta_f}(y)s\left(y\right)dy = \mathcal{E}s\left(y\right). \tag{15.15}$$

That is, the KL divergence from f_{θ_f} to f_{θ_0} is the expectation of LR when $\theta = \theta_f$. In the context of fault detection study, that means $D\left(f_{\theta_f}, f_{\theta_0}\right)$ is the expected value of the LR in case of faulty operations. Analogue to $D\left(f_{\theta_f}, f_{\theta_0}\right)$, the KL divergence from f_{θ_0} to f_{θ_f} is given by

$$D\left(f_{\theta_0}, f_{\theta_f}\right) = \int f_{\theta_0}(y)\log\frac{f_{\theta_0}(y)}{f_{\theta_f}(y)}dy = -\int f_{\theta_0}(y)s\left(y\right)dy = -\mathcal{E}s\left(y\right). \tag{15.16}$$

We would like to call reader's attention that the expected values $\mathcal{E}s\left(y\right)$ in (15.15) and (15.16) are different, and the one in (15.16) is achieved in case of fault-free operations. In order to distinguish these two different operation modes, which are of fundamental significance in fault detection, we denote them respectively by

$$D\left(f_{\theta_f}, f_{\theta_0}\right) = \mathcal{E}_{\theta_f}s\left(y\right), D\left(f_{\theta_0}, f_{\theta_f}\right) = -\mathcal{E}_{\theta_0}s\left(y\right).$$

15.2.2 KL Divergence Based Fault Detection

In this sub-section, we schematically introduce the basic idea and principle of KL divergence-based fault detection. Given the PDF $f_\theta(y)$ of the process measurement y with θ satisfying (15.1), our first task is to choose a test statistic between $D\left(f_{\theta_f}, f_{\theta_0}\right)$ and $D\left(f_{\theta_0}, f_{\theta_f}\right)$ due to the asymmetry of KL divergence. We suggest to use $D\left(f_{\theta_0}, f_{\theta_f}\right)$ instead of $D\left(f_{\theta_f}, f_{\theta_0}\right)$ for the following reasons:

- our objective is to solve the optimal fault detection problem formulated in Definition 2.4, which requires the determination of the threshold using the fault-free distribution or operation data,
- that is, for given α, the threshold is determined according to

$$\Pr\left(J > J_{th}\left|\theta = \theta_0\right.\right) = \alpha \Longleftrightarrow \Pr\left(J^{-1} < J_{th}^{-1}\left|\theta = \theta_0\right.\right) = \alpha, \tag{15.17}$$

- $D\left(f_{\theta_0}, f_{\theta_f}\right)$ is the expectation of the inversed LR during fault-free operation, and
- as a rule for the application of KL divergence from distribution P to Q, P typically represents the "true"distribution of data. This is, in our case, $\theta = \theta_0$.

Suppose that for the online fault detection, $y\left(i\right), i = 1, \cdots, n$, are collected. For the computation of $D\left(f_{\theta_0}, f_{\theta_f}\right)$ given in (15.16) using $y\left(i\right), i = 1, \cdots, n$, collected from the underlying distribution f_{θ_0}, we have to (i) estimate θ_f, since it is unknown, and (ii) approximate the expectation computation in (15.16). As a solution, we propose to use MLE of θ_f,

$$\hat{\theta}_f = \arg\max_{\theta_f} \sum_{i=1}^{n} \log L\left(\theta_f \left| y(i)\right.\right), \tag{15.18}$$

and the empirical expectation of $D\left(f_{\theta_0}, f_{\theta_f}\right)$,

$$\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) = \frac{1}{n}\left(\sum_{i=1}^{n} \log L\left(\theta_0 \left| y(i)\right.\right) - \sum_{i=1}^{n} \log L\left(\hat{\theta}_f \left| y(i)\right.\right)\right). \tag{15.19}$$

Recall that

$$D\left(f_{\theta_0}, f_{\theta_f}\right) = -\mathcal{E}s\left(y\right).$$

Hence, for the determination of the threshold satisfying (15.17) and on the use of $D\left(f_{\theta_0}, f_{\theta_f}\right)$ as the test statistic, we have

$$\begin{aligned} \Pr\left(J > J_{th} \left| \theta = \theta_0\right.\right) = \alpha &\Longleftrightarrow \Pr\left(D\left(f_{\theta_0}, f_{\theta_f}\right) < J_{th} \left| \theta = \theta_0\right.\right) = \alpha \\ &\Longleftrightarrow \Pr\left(D^{-1}\left(f_{\theta_0}, f_{\theta_f}\right) > J_{th}^{-1} \left| \theta = \theta_0\right.\right) = \alpha, \end{aligned} \tag{15.20}$$

and correspondingly the detection logic

$$\begin{cases} D_n\left(f_{\theta_0}, f_{\theta_f}\right) - J_{th} \geq 0, & \text{fault-free}, \\ D_n\left(f_{\theta_0}, f_{\theta_f}\right) - J_{th} < 0, & \text{faulty.} \end{cases}$$

Below is the algorithm running offline (in the training phase) for the threshold setting.

```
Algorithm 15.2 Step 0:  Set $\gamma = 0$;
Step 1:  For $j = 1$ to $N$
    Generate n data from the underlying distribution with the PDF $f_{\theta_0}(y)$, $y(i), i = 1, \cdots, n$, using the randomised algorithm;
    Calculate $\hat{\theta}_f$ and $\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)$ according to (15.18) and (15.19), respectively;
    If
        $\bar{D}_n^{-1}\left(f_{\theta_0}, f_{\theta_f}\right) > \gamma$    (15.21)
    then set
        $\gamma = \bar{D}_n^{-1}\left(f_{\theta_0}, f_{\theta_f}\right);$    (15.22)
    End.
Step 2:  Output $J_{th} = \gamma^{-1}$.
```

The number of iterations N is determined according to the following inequality

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\alpha}}, \tag{15.23}$$

where α is the FAR, $1-\delta$ is the confidence level with $\delta \in (0, 1)$. With (15.21)-(15.22) in Step 2 and N satisfying (15.23), we have, with a probability greater than $1 - \delta$, that

$$\Pr\left(D^{-1}\left(f_{\theta_0}, f_{\theta_f}\right) \leq J_{th}^{-1}\right) \geq 1 - \alpha \Longleftrightarrow \Pr\left(D^{-1}\left(f_{\theta_0}, f_{\theta_f}\right) > J_{th}^{-1}\right) \leq \alpha,$$

which, according to (15.20), results in

$$\Pr\left(D\left(f_{\theta_0}, f_{\theta_f}\right) < J_{th} \left|\theta = \theta_0\right.\right) \leq \alpha.$$

The background information and the proof of the above results will be given in the next chapter in our study on threshold setting using randomised algorithms technique.

It is clear that once J_{th} is set, the online fault detection can be performed using the following algorithm.

Algorithm 15.3 *KL divergence based online fault detection*

Step 1: *Collect data* $y(i), i = 1, \cdots, n$;
Step 2: *Compute* $\hat{\theta}_f$ *and* $\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)$ *according to (15.18) and (15.19), respectively;*
Step 3: *Check*

$$\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) - J_{th}$$

and run detection logic

$$\begin{cases} \bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) - J_{th} \geq 0, \textit{fault-free}, \\ \bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) - J_{th} < 0, \textit{faulty}. \end{cases}$$

15.2.3 KL Divergence and GLR Based Methods

Reviewing the published studies on applying KL divergence to fault detection shows that $D\left(f_{\theta_f}, f_{\theta_0}\right)$ has been commonly adopted as the test statistic for the detection purpose. According to (15.15), it can be understood as applying the expected value of the LR in case of faulty operations as the test statistic. This suggests that there should exist relations between the KL divergence and GLR. It is of considerable practical interests to reveal and understand these relations in the context of fault detection. Along the line in the study by Eguchi and Copas on interpreting KL divergence with Neyman-Pearson Lemma (the reader is referred to the reference given at the end of this chapter), we shall below investigate KL divergence and GLR based test statistics.

Consider $D\left(f_{\theta_f}, f_{\theta_0}\right)$ and re-write it into

$$D\left(f_{\theta_f}, f_{\theta_0}\right) = \int f_{\theta_f}(y) \log \frac{f_{\theta_f}(y)}{f_{\theta_0}(y)} dy = \mathcal{E}_{\theta_f} L_{\log}\left(\theta_f \left| y \right.\right) - \mathcal{E}_{\theta_f} L_{\log}\left(\theta_0 \left| y \right.\right),$$
$$\mathcal{E}_{\theta_f} L_{\log}\left(\theta_f \left| y \right.\right) = \int f_{\theta_f}(y) \log L\left(\theta_f \left| y \right.\right) dy,$$
$$\mathcal{E}_{\theta_f} L_{\log}\left(\theta_0 \left| y \right.\right) = \int f_{\theta_f}(y) \log L\left(\theta_0 \left| y \right.\right) dy.$$

In practice of fault detection, $\mathcal{E}_{\theta_f} L_{\log}\left(\theta_f \left| y \right.\right), \mathcal{E}_{\theta_f} L_{\log}\left(\theta_0 \left| y \right.\right)$ and so $D\left(f_{\theta_f}, f_{\theta_0}\right)$ will be replaced by their empirical realisations, analogue to our discussion in the last sub-section, as follows: for given n samples from the underlying distribution $f_{\theta_f}(y)$, $y(i), i = 1, \cdots, n$,

$$\begin{aligned}
&\bar{L}_{\log,\theta_f}\left(\theta_0 \left| y \right.\right) = \frac{1}{n} \sum_{i=1}^{n} L_{\log,\theta_f}\left(\theta_0 \left| y(i) \right.\right), \\
&\bar{L}_{\log,\theta_f}\left(\theta_f \left| y \right.\right) = \frac{1}{n} \sum_{i=1}^{n} L_{\log,\theta_f}\left(\theta_f \left| y(i) \right.\right) \Longrightarrow \\
&\bar{D}_n\left(f_{\theta_f}, f_{\theta_0}\right) = \bar{L}_{\log,\theta_f}\left(\theta_f \left| y \right.\right) - \bar{L}_{\log,\theta_f}\left(\theta_0 \left| y \right.\right) \\
&= \frac{1}{n}\left(\sum_{i=1}^{n} L_{\log,\theta_f}\left(\theta_f \left| y(i) \right.\right) - \sum_{i=1}^{n} L_{\log,\theta_f}\left(\theta_0 \left| y(i) \right.\right)\right),
\end{aligned}$$

where $\bar{L}_{\log,\theta_f}\left(\theta_0 \left| y \right.\right), \bar{L}_{\log,\theta_f}\left(\theta_f \left| y \right.\right)$ and $\bar{D}_n\left(f_{\theta_f}, f_{\theta_0}\right)$ denote the empirical realisations of $\mathcal{E}_{\theta_f} L_{\log}\left(\theta_f \left| y \right.\right), \mathcal{E}_{\theta_f} L_{\log}\left(\theta_0 \left| y \right.\right)$ and $D\left(f_{\theta_f}, f_{\theta_0}\right)$, respectively, and

$$L_{\log,\theta_f}\left(\theta_f \left| y(i) \right.\right) = \log L\left(\theta_f \left| y(i) \right.\right), L_{\log,\theta_f}\left(\theta_0 \left| y(i) \right.\right) = \log L\left(\theta_0 \left| y(i) \right.\right)$$

with $y(i)$ being generated from the distribution $f_{\theta_f}(y)$. Since θ_f is generally unknown and will be substituted by its MLE $\hat{\theta}_f$ in the fault detection practice, we finally have

$$\bar{D}_n\left(f_{\theta_f}, f_{\theta_0}\right) = \frac{1}{n}\left(\sum_{i=1}^{n} L_{\log,\theta_f}\left(\hat{\theta}_f \left| y(i) \right.\right) - \sum_{i=1}^{n} L_{\log,\theta_f}\left(\theta_0 \left| y(i) \right.\right)\right). \tag{15.24}$$

It seems formally that, apart from the factor $1/n$, $\bar{D}_n\left(f_{\theta_f}, f_{\theta_0}\right)$ given in (15.24) is identical with the GLR given in (15.6). This is, however, not true and inapplicable in the practice of fault detection.

Remember that the principle of applying GLR based test statistic for fault detection is to check if the test statistic during the *fault-free* operation is bounded by the threshold. And the threshold should be determined based on the fault-free distribution or the data collected during the fault-free operations. In other words, the samples $y(i), i = 1, \cdots, n$, used in (15.6) should be from the distribution $f_{\theta_0}(y)$. As discussed in the last sub-section, for the purpose of determining the threshold, the empirical realisation of $D\left(f_{\theta_0}, f_{\theta_f}\right)$,

$$\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) = -\frac{1}{n}\left(\sum_{i=1}^{n} L_{\log,\theta_0}\left(\hat{\theta}_f\left|y(i)\right.\right) - \sum_{i=1}^{n} L_{\log,\theta_0}\left(\theta_0\left|y(i)\right.\right)\right) \tag{15.25}$$
$$= \frac{1}{n}\sum_{i=1}^{n}\log\frac{L_{\theta_0}\left(\theta_0\left|y(i)\right.\right)}{L_{\theta_0}\left(\hat{\theta}_f\left|y(i)\right.\right)},$$

can be adopted. Here, the n samples, $y(i), i = 1, \cdots, n$, are from the underlying distribution $f_{\theta_0}(y)$, and being analogue and consistent to the above-introduced notations, and

$$\frac{1}{n}\sum_{i=1}^{n} L_{\log,\theta_0}\left(\theta_0\left|y(i)\right.\right), \frac{1}{n}\sum_{i=1}^{n} L_{\log,\theta_0}\left(\hat{\theta}_f\left|y(i)\right.\right),$$

are the empirical realisations of

$$\mathcal{E}_{\theta_0}L_{\log}\left(\theta_0\left|y\right.\right), \mathcal{E}_{\theta_0}L_{\log}\left(\hat{\theta}_f\left|y\right.\right),$$

respectively. It is evident that $-\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)$ is, apart from the factor $1/n$, equivalent with the GLR in (15.6). Since two test statistics deliver the identical fault detection performance, when their ratio is a constant, it can be concluded that the KL divergence $\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)$ and GLR based test statistics are equivalent with respect to fault detection performance. Note that, corresponding to $-\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)$ the threshold will be determined by

$$\Pr\left(\sum_{i=1}^{n}\log\frac{L_{\theta_0}\left(\theta_0\left|y(i)\right.\right)}{L_{\theta_0}\left(\hat{\theta}_f\left|y(i)\right.\right)} \leq J_{th}\right) = \alpha.$$

As a summary, we claim that $D\left(f_{\theta_0}, f_{\theta_f}\right)$ can be, equivalent to the GLR, applied to solving the fault detection problem formulated in Definition 2.4. In other words, the achieved solution leads to the maximum fault detectability for a given acceptable false alarm rate α.

Recall that in Definition 2.5, a dual optimal fault detection problem is formulated, in which the minimum false alarm rate would be reached for a given fault detection rate. That means, the threshold is first set for a given fault detection rate β, which is defined by

$$\Pr\left(J > J_{th}\left|\theta \neq \theta_0\right.\right) = \beta.$$

Thus, the threshold setting should be realised using the data from the underlying distribution $f_{\theta_f}(y)$. In this case, it is reasonable to use $D\left(f_{\theta_f}, f_{\theta_0}\right)$ as the test statistic.

15.3 Asymptotic Behaviour of GLR and KL Divergence as Test Statistics

The asymptotic behaviour of GLR and KL divergence has been well studied in the statistical research. In this section, we briefly summarise those existing results, which are useful for our fault detection work, without providing statistical descriptions and handlings in more details.

Recall that the online implementation of GLR and KL divergence based test statistics is performed using n (online) collected measurement data (samples). Corresponding to this, the threshold setting, for instance, performed in the offline training phase using Algorithm 15.1, should be realised on the assumption of n available samples. It is of considerable practical interests to know the statistical properties of the test statistic under consideration like its distribution, expectation or covariance, if sufficient number of data are available. That is, when n is sufficiently large. The statistical properties of the sample function for $n \rightarrow \infty$ are called asymptotic behaviour.

We first consider the likelihood function

$$L_{\log}\left(\theta_f \left| y\right.\right) = \sum_{i=1}^{n} L_{\log}\left(\theta_f \left| y\left(i\right)\right.\right).$$

Let

$$\hat{\theta}_f = \arg\max_{\theta_f} L_{\log}\left(\theta_f \left| y\right.\right)$$

be the MLE of θ_f using n samples which are generated (or collected) from the fault-free distribution $f_{\theta_0}(y)$. It is a well-known result in statistics that under certain trivial conditions the MLE almost surely converges to its true value as n i. i. d. samples approach to infinity. In our case,

$$\hat{\theta}_f \stackrel{n\rightarrow\infty}{\longrightarrow} \theta_0,$$

since the data are generated from the fault-free distribution $f_{\theta_0}(y)$. Moreover, $\hat{\theta}_f$ converges in distribution to a normal distribution satisfying

$$\sqrt{n}\left(\hat{\theta}_f - \theta_0\right) \longrightarrow \mathcal{N}\left(0, I_F^{-1}\right), \tag{15.26}$$

where I_F is the so-called Fisher information matrix whose (i, j) entry is given by

$$I_F\left(i, j\right) = -\mathcal{E}\left(\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log f_\theta(y\left|\theta\right.)\right), \theta = \theta_0.$$

We now consider $L_{\log}\left(\theta_f \left| y\right.\right)$ and its second order approximation using the Taylor series expansion at θ_0,

$$L_{\log}\left(\hat{\theta}_f\left|y\right.\right) \approx L_{\log}\left(\hat{\theta}_f\left|y\right.\right)\Big|_{\hat{\theta}_f=\theta_0} + \frac{\partial L_{\log}\left(\hat{\theta}_f\left|y\right.\right)}{\partial\hat{\theta}_f}\Big|_{\hat{\theta}_f=\theta_0}\left(\hat{\theta}_f-\theta_0\right)$$
$$+\frac{1}{2}\left(\hat{\theta}_f-\theta_0\right)^T G\left(\hat{\theta}_f-\theta_0\right),$$
$$G\left(i,j\right) = \frac{\partial^2}{\partial\theta_i\partial\theta_j}L_{\log}\left(\hat{\theta}_f\left|y\right.\right)\Big|_{\hat{\theta}_f=\theta_0}.$$

Since the MLE $\hat{\theta}_f$ reaches the maximum at θ_0, which yields

$$\frac{\partial L_{\log}\left(\hat{\theta}_f\left|y\right.\right)}{\partial\hat{\theta}_f}\Big|_{\hat{\theta}_f=\theta_0} = 0,$$

and $G\left(i,j\right)$ converges to, as $n \to \infty$,

$$\frac{1}{n}G\left(i,j\right) \to \mathcal{E}\left(\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log f_\theta(y\left|\theta\right.)\right),$$

we have, by noting (15.26),

$$L_{\log}\left(\theta_0\left|y\right.\right) - L_{\log}\left(\hat{\theta}_f\left|y\right.\right) \approx \frac{n}{2}\left(\hat{\theta}_f-\theta_0\right)^T I_F\left(\hat{\theta}_f-\theta_0\right) \Longrightarrow$$
$$2\left(L_{\log}\left(\theta_0\left|y\right.\right) - L_{\log}\left(\hat{\theta}_f\left|y\right.\right)\right) \sim \chi^2\left(\dim\left(\theta\right)\right).$$

In other words, the asymptotic behaviour of $s_n\left(y\right)$ can be described by χ^2-distribution with the degrees of freedom equal to the dimension of θ,

$$-2s_n\left(y\right) \sim \chi^2\left(\dim\left(\theta\right)\right). \tag{15.27}$$

Moreover, it follows from (15.25) that for $n \to \infty$

$$2n\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) \sim \chi^2\left(\dim\left(\theta\right)\right). \tag{15.28}$$

It is worth mentioning that (15.28) is coincident with the well-known result that

$$D\left(f_{\theta_0}, f_{\theta_f}\right) \approx \frac{1}{2}\left(\theta_f-\theta_0\right)^T I_F\left(\theta_f-\theta_0\right).$$

We would like to call reader's attention to the conditions that lead to the relation (15.27):

- $n \to \infty$,
- the second order approximation of $L_{\log}\left(\theta_f\left|y\right.\right)$ using its Taylor series expansion at θ_0.

The latter requires that θ_f should differ from θ_0 only very slightly. In the context of fault detection, this fact can be understood as a rule that in case of incipient faults, the distribution of the LR can be well approximated by χ^2 distribution. On the other hand, the requirement $n \to \infty$ makes the use of this nice result in practice more difficult. The well-known Hoeffding's inequality could help us to find a reasonable trade-off between the (limited) number n and the use of the asymptotic property (15.27) of the LR.

Lemma 15.1 *(Hoeffding's inequality) Given i. i. d. random variables* $x_1, \cdots, x_n$ *and* $x_i \in [a, b]$, *then for some* $\varepsilon > 0$

$$\Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_i - \mathcal{E}\frac{1}{n}\sum_{i=1}^{n} x_i\right| \geq \varepsilon\right) \leq 2e^{-\frac{2n\varepsilon^2}{(b-a)^2}}.$$

The Hoeffding's inequality gives a probabilistic relation between the empirical mean and the true mean value depending on the sample number. We now apply this result for our purpose. To be specific, we would like to find n so that

$$\Pr\left(\left|\frac{s_n(y)}{n} - \mathcal{E}\frac{s_n(y)}{n}\right| \geq \varepsilon\right) = \Pr\left(\left|\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right) - \mathcal{E}\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)\right| \geq \varepsilon\right) \leq \gamma. \tag{15.29}$$

Here, $s_n(y)$, $\bar{D}_n\left(f_{\theta_0}, f_{\theta_f}\right)$ are given in (15.6) and (15.19), respectively. $\varepsilon > 0$, $\gamma \in (0, 1)$ are some constants. Suppose that

$$s\left(y\left(i\right)\right) = \left(L_{\log}\left(\hat{\theta}_f \left|y\left(i\right)\right.\right) - L_{\log}\left(\theta_0 \left|y\left(i\right)\right.\right)\right) \in [a, b].$$

It is evident that (15.29) holds if

$$2e^{-\frac{2n\varepsilon^2}{(b-a)^2}} \leq \gamma,$$

which leads to

$$n \geq \frac{(b-a)^2}{2\varepsilon^2}\ln\frac{2}{\gamma}.$$

15.4 SPD Matrix Based Test Statistics and Fault Detection Schemes

In the previous sections, we have studied GLR and KL divergence based fault detection schemes. Roughly speaking, such schemes are established on the basis of known distributions or data-driven realisation of distributions of relevant random variables. In practice, due to uncertainties within and around the process under consideration, identifying a distribution to cover the overall (normal) operation of a process variable (possibly vectorised) is often a technical challenge that demands for considerable engineering efforts. On the other hand, it is often the case that during (the normal) operations, numerous data sets have been recorded, each of which or some of which as a group represent an operation mode under certain conditions. When we summarise these observations from the viewpoint of information geometry, the overall process operations can be abstracted as a manifold, and the distribution or the data representing an operation mode can be interpreted as a point in the manifold under consideration. To be specific, we denote an m-dimensional manifold by $\mathcal{M}$, a point in $\mathcal{M}$ with the local coordinate system by P_i and the geodesic distance between two points in $\mathcal{M}$, P_i and P_j, by $d\left(P_i, P_j\right)$. In this context, we propose the following fault detection schemes: given $P_i \in \mathcal{M}, i = 1, \cdots, n,$ which model the normal (fault-free) process operations,

- FD scheme I: find the mean of $P_i, i = 1, \cdots, n,$ denoted by P_M and defined by

$$P_M = \arg\min_P \sum_{i=1}^{n} d^2\left(P_i, P\right),$$

and set a threshold J_{th}, so that for any new point in $\mathcal{M}$, P_{new}, which is built during online process operation, the following decision logic is performed

$$\begin{cases} J_{th} - d^2\left(P_{new}, P_M\right) \geq 0 \Rightarrow \text{fault-free}, \\ J_{th} - d^2\left(P_{new}, P_M\right) < 0 \Rightarrow \text{faulty}; \end{cases}$$

- FD scheme II: Let $\xi \in \mathcal{R}$ be a measurement variable that represents, for instance, operation conditions. Suppose that for $i \in \{1, \cdots, n\}$, $\xi_i \in \mathcal{R}$ is associated with P_i. Let

$$P = M\left(P_\xi, P_0\right) \in \mathcal{M}, P_\xi, P_0 \in \mathcal{M},$$

model the normal operations with P_ξ, P_0 as model parameters to be identified. Here, P_ξ is a function of ξ. Find P_ξ, P_0 using data $\left(P_i, \xi_i\right), i = 1, \cdots, n,$ by solving the following optimisation problem

$$\left(\hat{P}_\xi, \hat{P}_0\right) = \arg\min_{P_\xi, P_0} \sum_{i=1}^{n} d^2\left(P_i, M\left(P_{\xi_i}, P_0\right)\right), \tag{15.30}$$

and set a threshold J_{th}, so that for any new point in $\mathcal{M}$ with the associated operation condition, $\left(P_{new}, \xi_{new}\right)$, the following decision logic is implemented

$$\begin{cases} J_{th} - d^2\left(P_{new}, M\left(\hat{P}_{\xi_{new}}, \hat{P}_0\right)\right) \geq 0 \Rightarrow \text{fault-free}, \\ \text{otherwise faulty}. \end{cases}$$

These two FD schemes can be applied for the detection purpose for processes running under different operational conditions. The first scheme is analogue to the commonly adopted fault detection strategy of detecting changes around the mean by taking into account uncertain variations in form of a threshold. The second scheme is in fact an extension of the first detection scheme. The idea behind that is, in addition to the constant matrix (parameterised as mean in the first FD scheme), to define an additional model parameter (matrix), which is a function of the operation conditions, so that the normal operation model can also reflect possible changes in the measurement data sets caused by the variation of the normal operation conditions. The challenge for the realisation of this FD scheme is the solution of the optimisation problem (15.30).

In the sequel, we will realise these ideas on the basis of measurement data in the format of symmetric and positive-definite (SPD) matrices.

15.4.1 Manifold of Symmetric and Positive-definite Matrices

Motivation In the framework of MVA, covariance matrices are commonly assumed to be SPD and used as a measurement of variations of a random vector around its (known) expectation (vector). In this context, the mean and covariance matrix of a vectorised random variable are of essential statistical importance for a successful fault detection. In practice, it is the nature of many industrial processes that process data are batchwise available. Let a batch data set be denoted by

$$Y = \left[y(1) \cdots y(l) \right] \in \mathcal{R}^{m \times l},$$

typically with $l >> m$. In industrial applications, due to the existing uncertainties it is often impossible to re-construct the expectation and covariance of a random variable from such a data set. Alternatively, we consider

$$P = \frac{1}{l} Y Y^T \in \mathcal{R}^{m \times m},$$

which can be interpreted as an approximation of the second moment of measurement vector y. In our subsequent work, we assume P is positive-definite and will study fault detection issues using measurement data in the format of SPD matrices.

All $m \times m$ dimensional SPDs form a $\frac{m(m+1)}{2}$ dimensional manifold, denoted by $\mathcal{P}(m)$. This allows to deal with our problems by applying existing differential-geometric methods. To this end, we first, in the sequel, briefly introduce some very basic differential-geometric properties of manifold $\mathcal{P}(m)$ as well as concepts and methods needed for our study. For details, the reader is referred to the references cited at the end of this chapter. We would like to emphasise that differential-geometric rather than statistical properties of SPD matrices lie in the focus of our subsequent investigation. Thus, no assumption is made on any statistical properties of our measurement variables.

Riemannian structure of $\mathcal{P}(m)$ Although for two matrices, $A \in \mathcal{R}^{m \times m}$, $B \in \mathcal{R}^{m \times m}$, the Frobenius-norm of $A - B$,

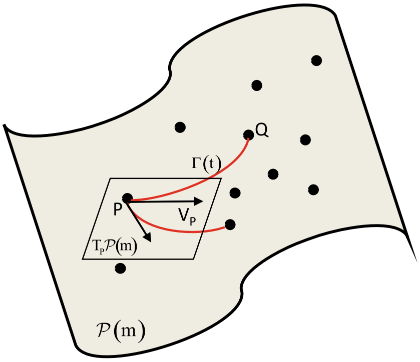


Fig. 15.1 Schematic description of some concepts in Riemannian manifold $\mathcal{P}\left(m\right)$

$$d_F(A, B) = \|A - B\|_F = \sqrt{tr\left((A - B)^T\,(A - B)\right)},$$

defines the Euclidean distance on the set of $m \times m$ real matrices, we are interested in $\mathcal{P}\left(m\right)$ as a special type of Riemannian manifolds and its geometric properties, like tangent space, geodesic curves, exponential and logarithmic maps and Riemannian distance, which are essential for our subsequent work. These concepts are schematically sketched in Fig. 15.1 and explained in detail below.

We denote the tangent space at $P \in \mathcal{P}\left(m\right)$ by $T_P\mathcal{P}\left(m\right)$, which is a $\frac{m(m+1)}{2}$ dimensional linear subspace, and call vectors in $T_P\mathcal{P}\left(m\right)$ tangent vectors at P. A tangent vector V_P belongs to $\mathcal{S}\left(m\right)$, where $\mathcal{S}\left(m\right)$ is the vector space of all $m \times m$ symmetric matrices, and can be interpreted as a directional derivative. Thus, in the context of our work on process monitoring, V_P represents (directional) variation at $P \in \mathcal{P}\left(m\right)$. On the tangent space at P, the inner product is defined as

$$\left\langle V_{P,1}, V_{P,2}\right\rangle_P = tr\left(P^{-1}V_{P,1}P^{-1}V_{P,2}\right), V_{P,1}, V_{P,2} \in T_P\mathcal{P}\left(m\right).$$

Correspondingly, the norm of $V_P \in T_P\mathcal{P}\left(m\right)$ is given by

$$\begin{aligned} \|V_P\|_P = \langle V_P, V_P\rangle_P^{1/2} = tr^{1/2}\left(P^{-1}V_P P^{-1}V_P\right) \\ = tr^{1/2}\left(P^{-1/2}V_P P^{-1/2}P^{-1/2}V_P P^{-1/2}\right) = \left\|P^{-1/2}V_P P^{-1/2}\right\|_F. \end{aligned} \tag{15.31}$$

This also leads to a natural definition of the Riemannian metric,

$$ds = \left\|P^{-1/2}(dP)P^{-1/2}\right\|_F,$$

which is the "infinitesimal length" at $P \in \mathcal{P}\left(m\right)$ as well. Equipped with the metric, Riemannian manifold $\mathcal{P}\left(m\right)$ is complete.

Let A be a regular $m \times m$ matrix. For

$$\hat{P} = A^T P A, P \in \mathcal{P}\left(m\right), \tag{15.32}$$

it holds

$$\begin{aligned} &\hat{P} \in \mathcal{P}(m), \\ &\quad \|V_P\|_{\hat{P}} = tr^{1/2}\left(A^{-1}P^{-1}A^{-T}V_P A^{-1}P^{-1}A^{-T}V_P\right) \\ &\qquad = tr^{1/2}\left(A^{-1}P^{-1}A^{-T}V_P A^{-1}P^{-1}A^{-T}V_P A^{-1}A\right) \\ &\qquad = tr^{1/2}\left(P^{-1}A^{-T}V_P A^{-1}P^{-1}A^{-T}V_P A^{-1}\right) = \left\|A^{-T}V_P A^{-1}\right\|_P, \end{aligned}$$

from which we have

$$ds = \left\|P^{-1/2}(dP)P^{-1/2}\right\|_F = \left\|A^{-1}P^{-1}A^{-T}A^T(dP)A\right\|_F.$$

That means, the Riemannian metric is invariant to the transformation (15.32).

Given $V_P \in T_P\mathcal{P}(m)$, $P \in \mathcal{P}(m)$, the exponential map at P, denoted by $\exp_P(V_P)$, maps the tangent vector V_P to $\mathcal{P}(m)$. The inverse of $\exp_P$ is called logarithmic map and denoted by $\log_P$. That is

$$\log_P : \mathcal{P}(m) \to T_P\mathcal{P}(m), \log_P\left(\exp_P(V_P)\right) = V_P.$$

Let $\Gamma(t)$ be the (unique) geodesic satisfying

$$\Gamma(0) = P, \dot{\Gamma}(0) = V_P.$$

The exponential map at P, $\exp_P(V_P)$, is defined as

$$\exp_P(V_P) = \Gamma(1).$$

It is proved that for Riemannian manifold $\mathcal{P}(m)$ with the tangent space $T_P\mathcal{P}(m)$,

$$\Gamma(t) = P^{1/2}(P^{-1/2}QP^{-1/2})^t P^{1/2}, Q \in \mathcal{P}(m), t \in [0,1], \tag{15.33}$$

is the geodesic. It is straightforward that

$$\Gamma(0) = P, \dot{\Gamma}(0) = P^{1/2}\log\left(P^{-1/2}QP^{-1/2}\right)P^{1/2} = V_P. \tag{15.34}$$

It becomes clear that

$$\exp_P(V_P) = P^{1/2}\exp\left(P^{-1/2}V_P P^{-1/2}\right)P^{1/2}, \tag{15.35}$$

$$\log_P(Q) = P^{1/2}\log\left(P^{-1/2}QP^{-1/2}\right)P^{1/2}, Q \in \mathcal{P}(m), \tag{15.36}$$

are Riemannian exponential and logarithmic maps, where $\log\left(P^{-1/2}QP^{-1/2}\right)$ and $\exp\left(P^{-1/2}V_P P^{-1/2}\right)$ are matrix logarithm and exponential, respectively. Note that

$$\exp_P(V_P) = P^{1/2} \exp\left(P^{-1/2} V_P P^{-1/2}\right) P^{1/2} = Q = \Gamma(1),$$

and thus the geodesic $\Gamma(t)$ can be written as

$$\Gamma(t) = P^{1/2}(P^{-1/2} Q P^{-1/2})^t P^{1/2} = P^{1/2} \exp\left(t P^{-1/2} V_P P^{-1/2}\right) P^{1/2}.$$

We would like to remark that for the complete Riemannian manifold $\mathcal{P}(m)$ the exponential map is defined for all tangent vectors at $P \in \mathcal{P}(m)$.

Remark 15.3 *In the above descriptions, the following definition of P^t, $P \in \mathcal{P}(m)$, $t \in \mathcal{R}$, is adopted*

$$P^t = e^{t \log P}, \tag{15.37}$$

which can be found in the textbooks on functions of matrices. The reader is referred to the references given in the last section of this chapter. It follows from (15.37) that

$$P^{t_1} P^{t_2} = e^{(t_1+t_2)\log P} = P^{(t_1+t_2)}, \log P^t = t \log P.$$

In general, geodesic distance on a manifold defines the shortest distance between two points in the manifold. For P_i and P_j in $\mathcal{P}(m)$, the geodesic distance, also known as Riemannian distance, is given by

$$d\left(P_i, P_j\right) = \left\| \log P_i^{-1/2} P_j P_i^{-1/2} \right\|_F. \tag{15.38}$$

By means of an SVD of $P_i^{-1/2} P_j P_i^{-1/2}$,

$$\begin{aligned} P_i^{-1/2} P_j P_i^{-1/2} &= U_{ij} \Sigma_{ij} U_{ij}^T, \\ \Sigma_{ij} &= diag\left(\lambda_1\left(P_i^{-1/2} P_j P_i^{-1/2}\right), \cdots, \lambda_m\left(P_i^{-1/2} P_j P_i^{-1/2}\right)\right) \end{aligned}$$

with $\lambda_k\left(P_i^{-1/2} P_j P_i^{-1/2}\right)$ being the k-th eigenvalue of matrix $P_i^{-1/2} P_j P_i^{-1/2}$, $k = 1, \cdots, m$, it turns out

$$\log P_i^{-1/2} P_j P_i^{-1/2} = U_{ij} \log \Sigma_{ij} U_{ij}^T.$$

It yields

$$d\left(P_i, P_j\right) = \left\| \log \Sigma_{ij} \right\|_F = \left(\sum_{k=1}^m \log^2 \lambda_k \left(P_i^{-1/2} P_j P_i^{-1/2}\right) \right)^{1/2}. \tag{15.39}$$

Since

$$\lambda_k\left(P_i^{-1/2}P_jP_i^{-1/2}\right)=\lambda_k\left(P_i^{-1/2}P_i^{-1/2}P_jP_i^{-1/2}P_i^{1/2}\right)=\lambda_k\left(P_i^{-1}P_j\right),$$
$$\lambda_k\left(P_i^{-1/2}P_jP_i^{-1/2}\right)=\lambda_k\left(P_i^{1/2}P_i^{-1/2}P_jP_i^{-1/2}P_i^{-1/2}\right)=\lambda_k\left(P_jP_i^{-1}\right),$$

it is evident that

$$d\left(P_j,P_i\right)=d\left(P_i,P_j\right)=d\left(P_i^{-1},P_j^{-1}\right),$$
$$d\left(P_i,P_j\right)=\left(\sum_{k=1}^{m}\log^2\lambda_k\left(P_i^{-1}P_j\right)\right)^{1/2}, \tag{15.40}$$

where $\lambda_k\left(P_i^{-1}P_j\right)$ denotes the k-th eigenvalue of matrix $P_i^{-1}P_j, k=1,\cdots,m.$

Next, we introduce the concept of Riemannian mean. Given n SPD matrices, $P_i, i=1,\cdots,n$, it can be easily proved that the arithmetic mean,

$$\bar{P}=\frac{1}{n}\sum_{i=1}^{n}P_i,$$

is the solution of the minimisation problem

$$\min_{\bar{P}}\sum_{i=1}^{n}d_F^2(P_i,\bar{P}).$$

That is, $\bar{P}$ is a point in $\mathcal{P}(m)$, which minimises the sum of its distances to the given point $P_i, i=1,\cdots,n$. In the same context, the Riemannian mean of $P_i, i=1,\cdots,n$, which is also called geometric mean, is defined by

$$P_g=\arg\min_{P}\sum_{i=1}^{n}d^2(P_i,P) \tag{15.41}$$

with $d(P_i,P)$ denoting the geodesic distance defined in (15.40).

Theorem 15.1 *The optimisation problem (15.41) is solvable if and only if*

$$\sum_{i=1}^{n}\log P_i^{-1}P_g=0. \tag{15.42}$$

This is a well-established result. The reader is referred to the references cited in the last section of this chapter. In general, the optimisation problem (15.41) cannot be solved in a closed-form. It has been reported in the literature cited at the end of this chapter that software aided numerical solutions based on the condition (15.42) are available.

15.4.2 Riemannian Distance Based Fault Detection Schemes

In this sub-section, we introduce numerous Riemannian distance based fault detection schemes as the realisations of the two fault detection schemes described at the beginning of this section.

Fault detection by checking variations around the Riemannian mean Suppose that, during the training phase, process data are recorded and formatted as $P_i \in \mathcal{P}(m), i = 1, \cdots, n$. Solving nonlinear equation (15.42) gives the Riemannian mean, P_g. For performing the online fault detection with new measurement data P_{new} using the detection logic,

$$J = d^2\left(P_g, P_{new}\right) \Longrightarrow \begin{cases} J_{th} - J \geq 0 \Rightarrow \text{fault-free}, \\ J_{th} - J < 0 \Rightarrow \text{faulty}, \end{cases} \tag{15.43}$$

the threshold J_{th} is to be determined. To this end, we propose four different schemes:

- Scheme I: The threshold is set to be

$$J_{th} := \max_{i \in \{1, \cdots, n\}} d^2\left(P_g, P_i\right); \tag{15.44}$$

- Scheme II: The threshold is set to be

$$J_{th} := \frac{1}{n} \sum_{i=1}^{n} d^2\left(P_g, P_i\right); \tag{15.45}$$

- Scheme III: Define n_0 so that

$$\frac{n_0}{n} \leq \alpha$$

  with α being the acceptable FAR. The threshold J_{th} is set so that the number of the data matrices, $P_j, j \in \{1, \cdots, n\}$, which lead to

$$d\left(P_g, P_j\right) > J_{th},$$

  is not larger than n_0;
- Scheme IV: Recall that the geodesic curve is parameterised by $t \in [0, 1]$ in (15.33). Hence,

$$\Gamma_g\left(t_i\left(j\right)\right) = P_g^{1/2}(P_g^{-1/2} P_i P_g^{-1/2})^{t_i(j)} P_g^{1/2}$$

  represents a point in the geodesic curve connecting P_g and P_i with $t_i\left(j\right) \in [0, 1]$ as a parameter. Note that

$$\Gamma_g\left(t_i\left(j\right)\right) \in \mathcal{P}(m), d\left(P_g, \Gamma_g\left(t_i\left(j\right)\right)\right) \leq d\left(P_g, P_i\right).$$

Thus, $\Gamma_g(t_i(j))$ is viewed as being in the range of normal (fault-free) operations. On this assumption, the following randomised algorithm is proposed for determining the threshold.

Algorithm 15.4 *A randomised algorithm aided threshold setting*

Step 0: *Determine N, the sampling number according to (15.23);*
Step 1: *Generate random samples*

$$t_i(j) \sim \mathcal{U}_{[0,1]}, i = 1, \cdots, n, j = 1, \cdots, N,$$

where $\mathcal{U}_{[0,1]}$ *denotes the uniform distribution in the interval* $[0, 1]$;
Step 2: *Compute, for* $i = 1, \cdots, n, j = 1, \cdots, N,$

$$d^2\left(P_g, \Gamma_g(t_i(j))\right) = \sum_{k=1}^{m} \log^2 \lambda_k\left((P_g^{-1/2} P_i P_g^{-1/2})^{t_i(j)}\right); \tag{15.46}$$

Step 3: *Determine the threshold, for instance, using threshold determination Scheme III given above.*

In the above algorithm, $t_i(j)$ (as parameter) is generated randomly using a randomised algorithm (see the next two chapters for details). As a result of (15.39) for the Riemannian distance computation, we finally have (15.46) for the Riemannian distance from P_g to $\Gamma_g(t_i(j))$.

On the assumption that the data set includes sufficient data, (threshold determination) Scheme I can be (very) conservative. This could be the case as well even if Schemes II–IV are adopted, when the measurement points (data) are less uniformly distributed in "directions"and "amplitudes". Inspired by the Mahalanobis distance (the χ^2-test statistic), we propose to use an SPD matrix to "concentrate"the data points. Remember that tangent vectors in $T_P\mathcal{P}(m)$ represent variations at point $P \in \mathcal{P}(m)$. The idea behind our effort is to concentrate the variations represented by the tangent vectors in $T_{P_g}\mathcal{P}(m)$. Let $V_{P_g}(i), i = 1, \cdots, n$, be the tangent vector at the mean P_g in direction P_i. It follows from (15.34) that

$$V_{P_g}(i) = P_g^{1/2} \log\left(P_g^{-1/2} P_i P_g^{-1/2}\right) P_g^{1/2}, i = 1, \cdots, n.$$

Corresponding to the norm of a tangent vector at $P \in \mathcal{P}(m)$ defined in (15.31), we introduce

$$\Sigma_g = \frac{1}{n}\sum_{i=1}^{n}\left(P_g^{-1/2} V_{P_g}(i) P_g^{-1/2}\right)^2 = \frac{1}{n}\sum_{i=1}^{n} \log^2\left(P_g^{-1/2} P_i P_g^{-1/2}\right) \in \mathcal{P}(m), \tag{15.47}$$

whose inverse is then adopted for concentrating the variations. As a result, we define the following evaluation function

$$\begin{aligned} J &= tr\left(P_g^{-1/2} V_{P_g}(P_{new}) P_g^{-1/2} \Sigma_g^{-1} P_g^{-1/2} V_{P_g}(P_{new}) P_g^{-1/2}\right) \\ &= tr\left(\log\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right) \Sigma_g^{-1} \log\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right)\right). \end{aligned} \tag{15.48}$$

Here, P_{new} is the new measurement data for the online fault detection. We would like to call reader's attention that for $\Sigma_g = I$,

$$J = tr\left(\log\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right) \log\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right)\right) = d^2\left(P_g, P_{new}\right).$$

In other words, $J^{1/2}$ defined in (15.48) with the concentrating matrix Σ_g^{-1} can be viewed as the distance with weighting. For the computation of J, we suggest the use of the following algorithm.

Algorithm 15.5 *A modified version with data concentration*

- *Offline computation (embedded into the training phase):*
  - *Do SVD*

$$P_g^{-1/2} P_i P_g^{-1/2} = U_i \Sigma_i U_i^T, \quad \Sigma_i = diag\left(\lambda_1\left(P_g^{-1/2} P_i P_g^{-1/2}\right), \cdots, \lambda_m\left(P_g^{-1/2} P_i P_g^{-1/2}\right)\right)$$

and calculate

$$\Sigma_g = \frac{1}{n} \sum_{i=1}^{n} U_i diag\left(\cdots, \log \lambda_k^2\left(P_g^{-1/2} P_i P_g^{-1/2}\right), \cdots\right) U_i^T;$$

  - *Do SVD*

$$\Sigma_g = U_g \bar{\Sigma}_g U_g^T, \ \bar{\Sigma}_g = diag\left(\lambda_1, \cdots, \lambda_m\right)$$

and save $U_g, \lambda_k, k = 1, \cdots, m$;

- *Online computation (for fault detection):*
  - *Do SVD*

$$P_g^{-1/2} P_{new} P_g^{-1/2} = U_{new} \Sigma_{new} U_{new}^T, \quad \Sigma_{new} = diag\left(\lambda_1\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right), \cdots, \lambda_m\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right)\right);$$

  - *Calculate*

$$J = tr\begin{pmatrix} diag\left(\cdots, \log \lambda_k^2\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right), \cdots\right) \cdot \\ \cdot U_{new}^T U_g diag\left(\lambda_1^{-1}, \cdots, \lambda_m^{-1}\right) U_g^T U_{new} \end{pmatrix}. \tag{15.49}$$

Equation (15.49) is the result of the following computation

$$\begin{aligned} J &= tr\left(\log\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right) \Sigma_g^{-1} \log\left(P_g^{-1/2} P_{new} P_g^{-1/2}\right)\right) \\ &= tr\left(U_{new} \log \Sigma_{new} U_{new}^T U_g diag\left(\lambda_1^{-1}, \cdots, \lambda_m^{-1}\right) U_g^T U_{new} \log \Sigma_{new} U_{new}^T\right) \\ &= tr\left(\log^2 \Sigma_{new} U_{new}^T U_g diag\left(\lambda_1^{-1}, \cdots, \lambda_m^{-1}\right) U_g^T U_{new}\right). \end{aligned}$$

In (15.49), $U_{new}^T U_g$ can be interpreted as "concentration" in directions and $diag\left(\lambda_1^{-1}, \cdots, \lambda_m^{-1}\right)$ as "concentration" in amplitudes.

The threshold setting corresponding to evaluation function J given in (15.48) or (15.49) can be realised analogue to the threshold determination schemes proposed at the beginning of this sub-section.

Fault detection using a simple model on Riemannian manifold $\mathcal{P}(m)$ The idea behind the fault detection scheme to be proposed in this sub-section is to identify a (simple) model embedded in Riemannian manifold $\mathcal{P}(m)$, which is parameterised by the operation conditions like speed, temperature etc. In a certain sense, this concept is similar to the linear parameter varying (LVP) paradigm known in control theory.

As cited at the end of this chapter, Fletcher has proposed a simple model based on the exponential map on Riemannian manifold in 2013, in which the tangent vector is multiplied by a (scalar) measurement variable representing the operation condition. We adopt this model for our purpose, since it well fits our needs and requirements, viewed from the following two aspects:

- it is a natural extension of the "mean model"introduced in the last sub-section to include variations in form of a tangent vector in the model,
- the possible variations caused by the changes in the operations are modelled by the tangent vector together with the (scalar) variable representing the operation condition.

For our purpose, the exponential map on $\mathcal{P}(m)$, as given in (15.35), is extended to

$$\hat{P}(\xi) = \exp_{P_M}\left(V_{P_M}(\xi)\right) = P_M^{1/2} \exp\left(P_M^{-1/2} V_{p_M}(\xi) P_M^{-1/2}\right) P_M^{1/2} \in \mathcal{P}(m) \tag{15.50}$$

with P_M being a point in $\mathcal{P}(m)$ and

$$V_{P_M}(\xi) = \xi V_{P_M}, V_{P_M} \in T_P \mathcal{P}(m), \xi \in \mathcal{R}.$$

The exponential map $\exp_{P_M}\left(V_{P_M}(\xi)\right)$ is the model to be identified, which delivers an estimate for a point in $\mathcal{P}(m)$ with respect to the operation condition described by the (measurement) variable ξ. To this end, the following optimisation problem will be solved: given data pairs, $(P_i, \xi_i), i = 1, \cdots, n$, here, $P_i \in \mathcal{P}(m)$ is the measurement data set and $\xi_i \in \mathcal{R}$ represents a certain operation condition, find P_M, V_{P_M} so that

$$C = \sum_{i=1}^{n} d^2\left(P_i, \hat{P}(\xi_i)\right) \tag{15.51}$$

is minimised. It is clear that

$$C = \sum_{i=1}^{n} tr\left(\log^2\left(P_i^{-1/2}\hat{P}\left(\xi_i\right)P_i^{-1/2}\right)\right).$$

Once the model (15.50) is successfully identified with $P_M \in \mathcal{P}(m), V_{P_M} \in T_P\mathcal{P}(m)$, we can apply the following algorithm for online fault detection:

Algorithm 15.6 *Operation condition depending fault detection*

- *Collect data and build P_{new} with the corresponding operation condition ξ_{new};*
- *Compute*

$$\hat{P}\left(\xi_{new}\right) = \exp_{P_M}\left(V_{P_M}\left(\xi_{new}\right)\right)$$

and further

$$J = d^2\left(P_{new}, \hat{P}\left(\xi_{new}\right)\right);$$

- *Run the detection logic*

$$\begin{cases} J_{th} - J \geq 0 \Rightarrow \text{fault-free}, \\ J_{th} - J < 0 \Rightarrow \text{faulty}. \end{cases}$$

Here, J_{th} can be set, for instance, equal to

$$J_{th} := \frac{1}{n}\sum_{i=1}^{n} d^2\left(P_i, \hat{P}\left(\xi_i\right)\right). \tag{15.52}$$

15.4.3 Clustering and Clustering Based Fault Detection Schemes

Although the operation conditions are taken into account in the simple model (15.50), its fault detection performance may be limited if the collected (fault-free operation) data are sparely distributed in the $\mathcal{P}(m)$ manifold, for instance, due to significantly different operation conditions. A good solution to this problem is to perform clustering of the process data before applying the fault detection algorithms proposed in the last sub-section. To this end, the well-established k-means strategy is adopted as follows.

For the simplicity, we assume that k (an integer) normal operation conditions are known and $P_i \in \mathcal{P}(m), i = 1, \cdots, n$, are collected for the clustering purpose.

Algorithm 15.7 *Riemannian distance based k-means algorithm clustering*

Step 0: Define k initial geometric means $P_{g,i}^0, i = 1, \cdots, k,$ with super index 0 *denoting the initial iteration. Set* $j = 0$;

Step 1: Build the clusters

$$C_i^{j+1} = \left\{ \begin{array}{c} P_j : d\left(P_j, P_{g,i}^j\right) \leq d\left(P_j, P_{g,l}^j\right), l \neq i, l = 1, \cdots, k, \\ j \in \{1, \cdots, n\} \end{array} \right\}$$

for $i = 1, \cdots, k$*;*

Step 2: If

$$\forall i \in \{1, \cdots, k\}, C_i^{j+1} = C_i^j,$$

stop and output

$$C_i = C_i^{j+1}, i = 1, \cdots, k,$$

otherwise

$$P_{g,i}^{j+1} = \arg\min_P \sum_{P_i \in C_i^{j+1}} d^2(P_i, P)$$

and $j = j + 1$*, go to Step 1.*

Once the clustering is successfully performed, a fault detection can be realised by means of the following detection logic: given a new measurement $P_{new} \in \mathcal{P}(m)$,

$$\forall i \in \{1, \cdots, k\}, J(P_{new}, C_i) \leq J_{th,i} \Longrightarrow \text{fault-free, otherwise faulty.}$$

Here, $J(P_{new}, C_i)$ represents one of the evaluation functions introduced in the last sub-section, (15.43) or (15.48), and formed using the SPD points (data set) in C_i.

Remark 15.4 *The above clustering based fault detection scheme can also be extended to dealing with fault isolation problems as far as measurement data during faulty operations could be collected.*

In practice, it could be the case that a certain cluster may include (very) limited number of data sets. As a consequence, the geometric mean defined in (15.41) is less representative and insufficient for a reliable clustering and fault detection. As a solution of this problem, we adopt the so-called convex model, which has been proposed recently for clustering on the SPD Riemannian manifold.

Recall that a geodesic curve connecting $P_1, P_2 \in \mathcal{P}(m)$ is expressed by

$$\Gamma_{12}(t) = P_1^{1/2}(P_1^{-1/2} P_2 P_1^{-1/2})^t P_1^{1/2} = P_1^{1/2} \exp\left(t P_1^{-1/2} V_{P_2} P_1^{-1/2}\right) P_1^{1/2}$$

for $t \in [0, 1]$, and it is well-known as well that

$$P_{g,12} = \arg\min_P \sum_{i=1}^{2} d^2(P_i, P) = P_1 \left(P_1^{-1} P_2\right)^{1/2} = P_2 \left(P_2^{-1} P_1\right)^{1/2} \quad (15.53)$$

is the geometric mean of P_1 and P_2. Using the relations given in the following lemma, it becomes clear that the geometric mean of P_1 and P_2 is the middle point of the geodesic curve connecting P_1, P_2.

Lemma 15.2 *Given $P_1, P_2 \in \mathcal{P}(m)$ and P satisfying*

$$\sum_{i=1}^{2} \log P_i^{-1} P = 0,$$

then the following relations hold

$$\begin{aligned} P &= P_1 \left(P_1^{-1} P_2\right)^{1/2} = P_2 \left(P_2^{-1} P_1\right)^{1/2} = \left(P_2 P_1^{-1}\right)^{1/2} P_1 = \left(P_1 P_2^{-1}\right)^{1/2} P_2 \\ &= P_1^{1/2} (P_1^{-1/2} P_2 P_1^{-1/2})^{1/2} P_1^{1/2} = P_2^{1/2} (P_2^{-1/2} P_1 P_2^{-1/2})^{1/2} P_2^{1/2}. \end{aligned} \tag{15.54}$$

The proof of this lemma can be found in the reference given at the end of this chapter.

It is of interest to notice that (15.53) can be equivalently written as

$$P_{g,12} = \arg\min_P \frac{1}{2} \sum_{i=1}^{2} d^2(P_i, P) = \arg\min_P \sum_{i=1}^{2} w_i d^2(P_i, P), w_1 = w_2 = \frac{1}{2}.$$

For a more general case, we first give the following lemma.

Lemma 15.3 *Given $P, Q \in \mathcal{P}(m)$ and*

$$G(Q) = \left\| \log P^{-1} Q \right\|_F^2, \tag{15.55}$$

then the gradient of $G(Q)$ is given by

$$\nabla G(Q) = \left(\log P^{-1} Q\right) P^{-1}. \tag{15.56}$$

The proof of this lemma is in fact a part of the proof of existence condition (15.42) given in Theorem 15.1, which can be found in the references cited at the end of this chapter.

We now in a position to give a more general problem formulation and its solution. Given $P_1, P_2 \in \mathcal{P}(m)$, find the weighted geometric mean as defined as

$$P_g = \arg\min_P \sum_{i=1}^{2} w_i d^2(P_i, P), w_1 + w_2 = 1, w_1, w_2 \geq 0. \tag{15.57}$$

For the sake of simplicity and without loss of generality, it is assumed that P_1 is achieved by a normalisation of the data sets. The solution of the optimisation problem (15.57) is described in the following theorem.

Theorem 15.2 *Given $P_1, P_2 \in \mathcal{P}(m)$, then for $w_1, w_2 \geq 0, w_1 + w_2 = 1$,*

$$P_g = \arg\min_P \sum_{i=1}^{2} w_i d^2(P_i, P) = P_1^{1/2}(P_1^{-1/2} P_2 P_1^{-1/2})^{w_2} P_1^{1/2}. \tag{15.58}$$

Proof Since

$$C(P) = \sum_{i=1}^{2} w_i d^2(P_i, P)$$

is a convex function, the optimisation problem (15.2) is solvable if and only if

$$\nabla \left(P_g\right) = 0,$$

which, following Lemma 15.3, is equivalent to

$$\sum_{i=1}^{2} w_i \log P_i^{-1} P_g = 0. \tag{15.59}$$

Note that

$$\sum_{i=1}^{2} w_i \log P_i^{-1} P_g = w_1 \log P_1^{1/2} P_g + w_2 \log P_2^{-1} P_g.$$

Substituting P_g in the above equation by $P_1^{1/2}(P_1^{-1/2} P_2 P_1^{-1/2})^{w_2} P_1^{1/2}$ given in (15.2) yields

$$\begin{aligned}
w_1 \log P_1^{-1} P_g &= w_1 \log P_1^{-1/2} P_g P_1^{1/2} = w_1 \log(P_1^{-1/2} P_2 P_1^{-1/2})^{w_2} \\
&= w_1 w_2 \log(P_1^{-1/2} P_2 P_1^{-1/2}), \\
w_2 \log P_2^{-1} P_g &= w_2 \log P_1^{1/2} P_2^{-1} P_1^{1/2} P_1^{-1/2} P_g P_1^{-1/2} \\
&= w_2 \log \left(P_1^{1/2} P_2^{-1} P_1^{1/2} (P_1^{-1/2} P_2 P_1^{-1/2})^{w_2} \right) \\
&= w_2 \log(P_1^{-1/2} P_2 P_1^{-1/2})^{-1+w_2} = -w_1 w_2 \log(P_1^{-1/2} P_2 P_1^{-1/2}) \\
&\Longrightarrow \sum_{i=1}^{2} w_i \log \left(P_i^{-1} P_g\right) = 0,
\end{aligned}$$

since $P_1^{-1/2} P_2 P_1^{-1/2} \in \mathcal{P}(m)$. Thus, according to (15.2)

$$P_g = P_1^{1/2}(P_1^{-1/2} P_2 P_1^{-1/2})^{w_2} P_1^{1/2}$$

is the solution of the optimisation problem (15.2).

Recall that $P_1^{1/2}(P_1^{-1/2} P_2 P_1^{-1/2})^{w_2} P_1^{1/2}$ is the point at

$$t = w_2$$

in the geodesic curve connecting P_1, P_2. In other words, finding a weighted geometric mean of two points in $\mathcal{P}(m)$ is equivalent to finding a point in the geodesic curve connecting these two points. For a cluster with n points, this formulation can be written as: given a cluster C with SPD points $P_i \in \mathcal{P}(m), i = 1, \cdots, n$, for any w_i, w_j satisfying

$$w_j = 1 - w_i, i \neq j, i, j \in \{1, \cdots, n\}, \sum_{k=1}^{n} w_k = 1, \tag{15.60}$$

the weighted geometric mean defined by

$$P_{g,ij} = \arg\min_P \sum_{k=1}^{n} w_k d^2(P_k, P) = \arg\min_P \left(w_i d^2(P_i, P) + w_j d^2(P_j, P)\right) \tag{15.61}$$

is a point in the geodesic curve connecting P_i and P_j. That means, the set with all SPD points in the geodesic curve connecting any two points in C can be equivalently expressed by the solution of the optimisation problem given in (15.61) with different weighting factors satisfying (15.60). We now remove the assumption $w_j = 1 - w_i$ and consider a more general case: let

$$P_g = \arg\min_P \sum_{i=1}^{n} w_i d^2(P_i, P), \sum_{i=1}^{n} w_i = 1$$

be the weighted mean of $P_i \in \mathcal{P}(m), i = 1, \cdots, n$, for given weighting factors $\{w_1 \geq 0, \cdots, w_n \geq 0\}$. We call the set

$$\mathcal{P}_g = \left\{ P_g : \forall w_{\acute{\imath}} \geq 0, i = 1, \cdots, n, \sum_{i=1}^{n} w_i = 1, P_g = \arg\min_P \sum_{i=1}^{n} w_i d^2(P_i, P) \right\} \tag{15.62}$$

the convex set of the cluster C and use it to represent operations for the cluster C. In other words, for a given cluster C with (very) limited number of measurement data sets, $P_i, i = 1, \cdots, n$, the multivariate interpolations based on the support points $P_i, i = 1, \cdots, n$, result in additional (many) points in C, which would be helpful for a successful fault detection. Note that $\mathcal{P}_g$ is indeed a sub-manifold in $\mathcal{P}(m)$.

To check if a measurement data set belongs to the cluster C and moreover to distinguish two clusters, we introduce below the definitions of (i) distance of a point in $\mathcal{P}(m)$ to the sub-manifold $\mathcal{P}_g$ in $\mathcal{P}(m)$, (ii) distance between sub-manifold $\mathcal{P}_{1,g}, \mathcal{P}_{2,g}$ in $\mathcal{P}(m)$.

Definition 15.1 *Given $P \in \mathcal{P}(m)$ and $\mathcal{P}_g \subset \mathcal{P}(m)$ defined by (15.62), the distance from P to $\mathcal{P}_g$, $d\left(P, \mathcal{P}_g\right)$, is defined by*

$$d\left(P,\mathcal{P}_g\right)=\min_{P_g\in\mathcal{P}_g} d\left(P,P_g\right). \tag{15.63}$$

Moreover, given $\mathcal{P}_{1,g},\mathcal{P}_{2,g}\subset\mathcal{P}(m)$ *with* $\mathcal{P}_{i,g}, i=1,2,$ *being defined by (15.62), the distance between* $\mathcal{P}_{1,g}$ *and* $\mathcal{P}_{2,g}$, $d\left(\mathcal{P}_{1,g},\mathcal{P}_{2,g}\right)$, *is defined by*

$$d\left(\mathcal{P}_{1,g},\mathcal{P}_{2,g}\right)=\min_{P_i\in\mathcal{P}_{1,g},P_j\in\mathcal{P}_{2,g}} d\left(P_i,P_j\right). \tag{15.64}$$

Remark 15.5 *The above definitions are standard for the distance between a point (in a subspace/manifold) to a set/sub-manifold as well as the distance between two sets/sub-manifolds.*

Recall that any point in $\mathcal{P}_g$ is the weighted geometric mean of $P_i, i=1,\cdots,n,$ for given $w_i\geq 0, i=1,\cdots,n,$ satisfying $\sum_{i=1}^{n} w_i=1$. Moreover, according to Lemma 15.3, a necessary and sufficient condition for

$$P_g=\arg\min_{P}\sum_{i=1}^{n} w_i d^2(P_i,P)$$

is

$$\sum_{i=1}^{n} w_i\log\left(P_i^{-1}P_g\right)=0.$$

Thus, the minimisation problem (15.63) can be also equivalently written as

$$\begin{aligned} d\left(P,\mathcal{P}_g\right)&=\min_{w_i\geq 0,i=1,\cdots,n} d\left(P,P_g\right), \\ \text{s.t. (i)}\ &\sum_{i=1}^{n} w_i\log\left(P_i^{-1}P_g\right)=0, \\ \text{(ii)}\ &\sum_{i=1}^{n} w_i=1. \end{aligned} \tag{15.65}$$

In terms of the distance defined above, we are now in the position to perform clustering or fault detection as follows: given a new measurement $P_{new}\in\mathcal{P}(m)$,

$$d\left(P_{new},\mathcal{P}_g\right)\leq J_{th}\Longrightarrow P_{new}\in C,$$

otherwise P_{new} belongs to another cluster or faulty.

Here, J_{th} is the threshold and can be determined, for instance, using randomised algorithms technique.

In order to evidently distinguish two clusters, C_1 and C_2, represented by their convex sets, $\mathcal{P}_{1,g},\mathcal{P}_{2,g}\subset\mathcal{P}(m)$ with support points, $P_{1,i}\in\mathcal{P}_{1,g}$, $P_{2,j}\in\mathcal{P}_{2,g}$, we suggest to apply the following alternating minimisation based iterative algorithm proposed by Amari (see the reference given in the next section).

We begin with an arbitrary point $P_{1,0}$ in $\mathcal{P}_{1,g}$ and solve the optimisation problem (15.65) with

$$d\left(P_{1,0}, \mathcal{P}_{2,g}\right) = \min_{w_i \geq 0, i=1,\cdots,n} d\left(P_{1,0}, P_{2,g}\right),$$

$$\text{s.t. (i)} \sum_{i=1}^{n} w_i \log\left(P_{2,i}^{-1} P_{2,g}\right) = 0, \text{ (ii)} \sum_{i=1}^{n} w_i = 1, P_{2,i} \in C_2.$$

Suppose that the solution of the above optimisation problem is

$$P_{2,1} = \arg \min_{w_i \geq 0, i=1,\cdots,n} d\left(P_{1,0}, P_{2,g}\right) \in \mathcal{P}_{2,g}.$$

Now, solve the optimisation problem (15.65) with

$$d\left(P_{2,1}, \mathcal{P}_{1,g}\right) = \min_{w_i \geq 0, i=1,\cdots,n} d\left(P_{2,1}, P_{1,g}\right),$$

$$\text{s.t. (i)} \sum_{i=1}^{n} w_i \log\left(P_{1,i}^{-1} P_{1,g}\right) = 0, \text{ (ii)} \sum_{i=1}^{n} w_i = 1, P_{1,i} \in C_1,$$

which delivers

$$P_{1,1} = \arg \min_{w_i \geq 0, i=1,\cdots,n} d\left(P_{2,1}, P_{1,g}\right) \in \mathcal{P}_{1,g}.$$

It is evident that

$$d\left(P_{2,1}, P_{1,1}\right) \leq d\left(P_{2,1}, P_{1,0}\right).$$

In general, we have

$$d\left(P_{2,i+1}, P_{1,i}\right) \leq d\left(P_{2,i}, P_{1,i}\right) \leq d\left(P_{2,i}, P_{1,i-1}\right).$$

Thus, repeating the above computations iteratively will lead to a converging solution.

15.4.4 Examples

Next, we consider two special types of fault detection problems to illustrate the theoretical results and fault detection schemes presented in the previous sub-sections.

Example 1 Consider $P_i \in \mathcal{P}(m), i = 1, \cdots, n$, collected during the fault-free operations. Let

$$P_i = U\Sigma_i U^T,\, \Sigma_i = diag\left(\sigma_{i1}, \cdots, \sigma_{im}\right), i = 1, \cdots, n,$$

be the SVD of P_i. Suppose that

$$\Sigma_i \neq \Sigma_j, i \neq j, i, j = 1, \cdots, n,$$

which means, operation uncertainties do not cause changes in the directions expressed by U (consisting of the eigen-vectors) but may lead to variations of the length of some directions (expressed by the singular values).

To determine the geometric mean of $P_i, i = 1, \cdots, n$, we use the relation

$$\sum_{i=1}^{n} \log P_i^{-1} P_g = \sum_{i=1}^{n} \log \Sigma_i^{-1} U P_g U^T$$

and set

$$P_g = U^T \Sigma_g U,\, \Sigma_g = diag\left(\sigma_{g1}, \cdots, \sigma_{gm}\right),$$

which yields

$$\begin{aligned}\sum_{i=1}^{n} \log P_i^{-1} P_g &= \sum_{i=1}^{n} \log diag\left(\sigma_{i1}^{-1}\sigma_{g1}, \cdots, \sigma_{im}^{-1}\sigma_{gm}\right) \\ &= diag\left(\log \sigma_{g1}^{n} \prod_{i=1}^{n} \sigma_{i1}^{-1}, \cdots, \log \sigma_{gm}^{n} \prod_{i=1}^{n} \sigma_{im}^{-1}\right).\end{aligned}$$

As a result, it becomes clear that

$$\sigma_{gj}^{n} = \prod_{i=1}^{n} \sigma_{ij},\, j = 1, \cdots, m,\, P_g = U^T \Sigma_g U \tag{15.66}$$

solve the equation

$$\sum_{i=1}^{n} \log P_i^{-1} P_g = 0$$

and so the optimisation problem (15.41).

Based on the solution (15.66), we are able, for instance, to set threshold (15.44) defined in Scheme I equal to

$$\begin{aligned}J_{th} &= \max_{i \in \{1, \cdots, n\}} d^2\left(P_g, P_i\right) = \max_{i \in \{1, \cdots, n\}} \left\| \log P_i^{-1} P_g \right\|_F^2 \\ &= \max_{i \in \{1, \cdots, n\}} \sum_{j=1}^{m} \left(\frac{1}{n} \sum_{k=1}^{n} \log \sigma_{kj} - \log \sigma_{ij}\right)^2\end{aligned}$$

or threshold (15.45) defined in Scheme II equal to

$$\begin{aligned} J_{th} &= \frac{1}{n}\sum_{i=1}^{n} d^2\left(P_g, P_i\right) = \frac{1}{n}\sum_{i=1}^{n} \left\| \log P_i^{-1} P_g \right\|_F^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{k=1}^{n}\log\sigma_{kj} - \log\sigma_{ij}\right)^2 . \end{aligned}$$

Next, we derive the convex set built by $P_i \in \mathcal{P}(m), i = 1, \cdots, n$, according to (15.62). Recall that for given $w_{\acute{i}} \geq 0, i = 1, \cdots, n$, and $\sum_{i=1}^{n} w_i = 1$,

$$P_g = \arg\min_{P}\sum_{i=1}^{n} w_i d^2(P_i, P) \Longleftrightarrow \sum_{i=1}^{n} w_i \log\left(P_i^{-1}P_g\right) = 0.$$

Since for $P_g = U^T \Sigma_g U$,

$$\begin{aligned} \sum_{i=1}^{n} w_i \log P_i^{-1} P_g &= \sum_{i=1}^{n} \log diag\left(\sigma_{i1}^{-w_i}\sigma_{g1}^{w_i}, \cdots, \sigma_{im}^{-w_i}\sigma_{gm}^{w_i}\right) \\ &= diag\left(\log\sigma_{g1}^{\sum_{i=1}^{n} w_i}\prod_{i=1}^{n}\sigma_{i1}^{-w_i}, \cdots, \log\sigma_{gm}^{\sum_{i=1}^{n} w_i}\prod_{i=1}^{n}\sigma_{im}^{-w_i}\right), \end{aligned}$$

it turns out

$$\sigma_{gj} = \prod_{i=1}^{n}\sigma_{ij}^{w_i}, j = 1, \cdots, m.$$

Hence,

$$\mathcal{P}_g = \left\{ \begin{array}{c} P_g = U^T diag\left(\cdots, \prod_{i=1}^{n}\sigma_{ij}^{w_i}, \cdots\right) U : \\ \forall w_{\acute{i}} \geq 0, i = 1, \cdots, n, \sum_{i=1}^{n} w_i = 1 \end{array} \right\},$$

and it becomes evident that any point in $\mathcal{P}_g$ is a function of $w_{\acute{i}} \geq 0, i = 1, \cdots, n$.

Example 2 We now consider two clusters of data sets, C_1 and C_2,

$$C_1 = \left\{P_{1i} \in \mathcal{P}(m), i = 1, \cdots, n_1\right\}, C_2 = \left\{P_{2i} \in \mathcal{P}(m), i = 1, \cdots, n_2\right\}.$$

It is assumed that

$$\begin{aligned} P_{1i} &= U_1 \Sigma_{1i} U_1^T, \Sigma_{1i} = diag\left(\sigma_{1,i1}, \cdots, \sigma_{1,im}\right), i = 1, \cdots, n_1, \\ P_{2i} &= U_2 \Sigma_{2i} U_2^T, \Sigma_{2i} = diag\left(\sigma_{2,i1}, \cdots, \sigma_{2,im}\right), i = 1, \cdots, n_2. \end{aligned}$$

That is, both clusters are of the property of the data sets studied in Example 1. In order to get a deep insight into the problem formulations and the associated solutions, we suppose the $m \times m$ dimensional unit matrix, I, is the reference point in $\mathcal{P}(m)$, and the points in C_1 and C_2, $P_{1i}, P_{2j} \in \mathcal{P}(m), i = 1, \cdots, n_1, j = 1, \cdots, n_2$, are connected with I by the geodesic curves expressed by

$$\Gamma_{1i}(t) = P_{1i}^t, t \in [0, 1], i = 1, \cdots, n_1,$$
$$\Gamma_{2i}(t) = P_{2i}^t, t \in [0, 1], i = 1, \cdots, n_2.$$

Recall that a tangent vector is associated with each geodesic curve and given by

$$V_{P_{1i}} = \dot{\Gamma}_{1i}(0) = \log P_{1i}, i = 1, \cdots, n_1,$$
$$V_{P_{2i}} = \dot{\Gamma}_{2i}(0) = \log P_{2i}, i = 1, \cdots, n_2.$$

In the sequel, we consider an extreme case that

$$\Sigma_{1i} > I, \Sigma_{2j} < I, i = 1, \cdots, n_1, j = 1, \cdots, n_2. \tag{15.67}$$

Since

$$V_{P_{1i}} = \log P_{1i}, V_{P_{2i}} = \log P_{2i},$$

the assumption means, $\Gamma_{1i}(t)$ and $\Gamma_{2i}(t)$ move in (totally) different directions. This should allow an evident clustering between the two clusters, which will be studied below.

Remember that the geometric mean of C_1, for instance, is

$$P_{1g} = U_1^T \Sigma_{1g} U_1, \Sigma_{1g} = diag\left(\cdots, \left(\prod_{i=1}^{n_1} \sigma_{1ij}\right)^{1/n_1}, \cdots\right)$$

and it holds

$$\begin{aligned} J_{th,1} &= \max_{i \in \{1,\cdots,n_1\}} d^2\left(P_{1g}, P_{1i}\right) = \max_{i \in \{1,\cdots,n_1\}} \sum_{j=1}^{m} \left(\frac{1}{n_1} \sum_{k=1}^{n_1} \log \sigma_{1kj} - \log \sigma_{1ij}\right)^2 \\ &= \sum_{j=1}^{m} \left(\frac{1}{n_1} \sum_{k=1}^{n_1} \log \sigma_{1kj} - \log \sigma_{1qj}\right)^2, \\ P_{1q} &:= \arg \max_{i \in \{1,\cdots,n_1\}} d^2\left(P_{1g}, P_{1i}\right). \end{aligned}$$

We now check the distance between P_{1g} and any point in C_2, $P_{2i} \in C_2$,

$$d^2\left(P_{1g}, P_{2i}\right) = \left\|\log P_{2i}^{-1} P_{1g}\right\|_F^2 = \left\|\log P_{1g}^{1/2} P_{2i}^{-1} P_{1g}^{1/2}\right\|_F^2.$$

Notice that

$$\begin{aligned} P_{2i}^{-1} - \frac{1}{\bar{\sigma}_{2,i}} I \geq 0, \bar{\sigma}_{2,i} &= \max\left\{\sigma_{2,i1}, \cdots, \sigma_{2,im}\right\} \\ &\Longrightarrow \log P_{1g}^{1/2} P_{2i}^{-1} P_{1g}^{1/2} \geq \log\left(\frac{1}{\bar{\sigma}_{2,i}} P_{1g}^{1/2} P_{1g}^{1/2}\right) \\ &= diag\left(\cdots, \frac{1}{n_1}\sum_{k=1}^{n_1} \log \sigma_{1kj} - \log \bar{\sigma}_{2,i}, \cdots\right) \\ &\Longrightarrow d^2\left(P_{1g}, P_{2i}\right) \geq \sum_{j=1}^{m}\left(\frac{1}{n_1}\sum_{k=1}^{n_1} \log \sigma_{1kj} - \log \bar{\sigma}_{2,i}\right)^2 . \end{aligned}$$

As a result, due to equation (15.67) that leads to

$$\sum_{k=1}^{n_1} \log \sigma_{1kj} > 0, -\log \bar{\sigma}_{2,i} > 0, -\log \sigma_{1ij} < 0,$$

it turns out

$$\begin{aligned} d^2\left(P_{1g}, P_{2i}\right) &\geq \sum_{j=1}^{m}\left(\frac{1}{n_1}\sum_{k=1}^{n_1} \log \sigma_{1kj} - \log \bar{\sigma}_{2,i}\right)^2 > \sum_{j=1}^{m}\left(\frac{1}{n_1}\sum_{k=1}^{n_1} \log \sigma_{1kj}\right)^2 \\ &> \sum_{j=1}^{m}\left(\frac{1}{n_1}\sum_{k=1}^{n_1} \log \sigma_{1kj} - \log \sigma_{1qj}\right)^2 = J_{th,1}. \end{aligned} \tag{15.68}$$

The inequality (15.62) tells us, any point in C_2 will be correctly clustered to C_2 using the detection scheme introduced in the previous sub-sections.

We now extend this result to the convex sets of the clusters C_1 and C_2 defined by

$$\mathcal{P}_{1g} = \left\{\begin{array}{c} P_{1g} : \forall w_{\acute{\imath}} \geq 0, i = 1, \cdots, n_1, \sum_{i=1}^{n_1} w_i = 1, \\ P_{1g} = \arg\min_P \sum_{i=1}^{n_1} w_i d^2(P_{1i}, P), P_{1i} \in C_1 \end{array}\right\},$$

$$\mathcal{P}_{2g} = \left\{\begin{array}{c} P_{2g} : \forall w_{\acute{\imath}} \geq 0, i = 1, \cdots, n_2, \sum_{i=1}^{n_2} w_i = 1, \\ P_{2g} = \arg\min_P \sum_{i=1}^{n_2} w_i d^2(P_{2i}, P), P_{2i} \in C_2 \end{array}\right\}.$$

Using the results achieved in Example 1 yields

$$\mathcal{P}_{1g} = \left\{ \begin{array}{c} P_{1g} = U_1^T diag\left(\cdots, \prod_{i=1}^{n_1} \sigma_{1ij}^{w_i}, \cdots\right) U_1 : \\ \forall w_i \geq 0, i = 1, \cdots, n_1, \sum_{i=1}^{n_1} w_i = 1 \end{array} \right\},$$

$$\mathcal{P}_{2g} = \left\{ \begin{array}{c} P_{2g} = U_2^T diag\left(\cdots, \prod_{i=1}^{n_2} \sigma_{2ij}^{\beta_i}, \cdots\right) U_2 : \\ \forall \beta_i \geq 0, i = 1, \cdots, n_2, \sum_{i=1}^{n_2} \beta_i = 1 \end{array} \right\}.$$

We now check the distance of a point $P_{2g} \in \mathcal{P}_{2g}$ to the sub-manifold $\mathcal{P}_{1g}$, as given in Definition 15.1. To this end, consider the distance between any two $P_{1g} \in \mathcal{P}_{1g}$ and $P_{2g} \in \mathcal{P}_{2g}$:

$$\begin{aligned} d^2\left(P_{1g}, P_{2g}\right) &= \left\| \log P_{1g}^{1/2} P_{2g}^{-1} P_{1g}^{1/2} \right\|_F^2 \\ &= \left\| \log P_{1g}^{1/2} U_2^T diag\left(\cdots, \prod_{i=1}^{n_2} \sigma_{2ij}^{\beta_i}, \cdots\right)^{-1} U_2 P_{1g}^{1/2} \right\|_F^2 . \end{aligned}$$

Let

$$\gamma = \max\left\{ \prod_{i=1}^{n_2} \sigma_{2i1}^{\beta_i}, \cdots, \prod_{i=1}^{n_2} \sigma_{2im}^{\beta_i} \right\}.$$

It yields

$$d^2\left(P_{1g}, P_{2g}\right) \geq \left\| \log P_{1g}/\gamma \right\|_F^2 = \sum_{j=1}^{m} \left(\log \prod_{i=1}^{n_1} \sigma_{1ij}^{w_i} - \log\gamma \right)^2 .$$

Recall that

$$\begin{gathered} \forall i \in \{1, \cdots, n_1\}, j \in \{1, \cdots, m\}, \sigma_{1ij} > 1 \Longrightarrow \prod_{i=1}^{n_1} \sigma_{1ij}^{w_i} > 1, \\ \forall i \in \{1, \cdots, n_2\}, j \in \{1, \cdots, m\}, \sigma_{2ij} < 1 \Longrightarrow \gamma < 1 \Longrightarrow \\ d^2\left(P_{1g}, P_{2g}\right) \geq \sum_{j=1}^{m} \left(\log \prod_{i=1}^{n_1} \sigma_{1ij}^{w_i} - \log\gamma \right)^2 > 0. \end{gathered} \tag{15.69}$$

Since inequality (15.69) is true for any point in $\mathcal{P}_{1g}$, it can be concluded that

$$d\left(P_{2g}, \mathcal{P}_{1g}\right) = \min_{P_{1g} \in \mathcal{P}_{1g}} d\left(P_{2g}, P_{1g}\right) > 0.$$

This demonstrates a clear and unique clustering, as expected.

Example 3 Our third example deals with an application of the proposed Riemannian distance based fault detection schemes to performance monitoring of dynamic feedback control systems. In Chaps. 21 and 22, issues of performance monitoring and degradation detection for dynamic systems are addressed and numerous algorithms are proposed. Roughly speaking, the core of these schemes is the prediction and computation of the cost function

$$J(i)=\sum_{k=i}^{\infty}\left(y^T(k)Q_y y(k)+u^T(k)Q_u u(k)\right),\, Q_y\geq 0,\, Q_u>0,$$

using the online process data. On the assumption of a state feedback controller $u(k)=Fx(k)$, the value of $J(i)$ can be expressed by

$$J(i)=x^T(i)Px(i).$$

Here, $x(i)$ is the vector of the process state variables and P is an SPD matrix and identified using process data (the reader is referred to Chaps. 21 and 22 for different forms of the algorithms and solutions). The idea behind these schemes is utilisation of the fact that changes in the system dynamics will cause changes in P matrix and so in turn in $J(i)$. As well-known in control theory, SPD matrix P is in fact the solution of Lyapunov equation

$$P=A_F^T P A_F+Q,\, A_F=A+BF, \tag{15.70}$$

for discrete-time LTI systems or

$$A_F^T P+PA_F=-Q. \tag{15.71}$$

for continous-time systems. Here, it is assumed that the system under consideration has the minimal realisation (A, B, C) with A representing the system matrix, and $Q>0$ is often a function of B or C.

The following known results motivate us to apply Riemannian metric of P as a detection evaluation function to detect performance degradation:

- as a solution of Lyapunov equation, P can be expressed either by

$$P=\sum_{k=i}^{\infty}\left(A_F^k\right)^T QA_F^k$$

for Lyapunov equation (15.70) or by

$$P=\int_{t_0}^{\infty} e^{A_F^T\tau}Qe^{A_F\tau}d\tau$$

for Lyapunov equation (15.71),

- the system matrix A_F can be parameterised by P

$$A_F = P^{-1/2} U (P - Q)^{1/2}$$

for discrete-time LTI systems or

$$A_F^T = -\frac{1}{2} Q P^{-1} + V P^{-1}$$

for continous-time systems, where U, V are unitary and skey-symmetric matrix, respectively.

In other words, changes in P do reflect variations in the system dynamics.

For the performance-based detection purpose, we propose the following two schemes:

- data-driven scheme:
  - collect process data under different operation conditions;
  - identify $P_{i,} i = 1, \cdots, N$, using the algorithms given in Chaps. 21 and 22;
  - realise performance-based fault detection or isolation or clustering on the basis of $P_{i,} i = 1, \cdots, N$, using the algorithms proposed in this section.
- model-based detection scheme:
  - generate N random samples using RA on the basis of the (system) uncertainty model (see Chap. 17);
  - compute the corresponding $P_{i,} i = 1, \cdots, N$;
  - determine P_g and the threshold, for instance, using Algorithm 15.4;
  - For online detection: (i) identify P (see the algorithms given in Chaps. 21 and 22), and (ii) compute the Riemannian distance between P and P_g, and run detection logic.

15.5 Notes and References

χ^2- or T^2-test statistics are commonly adopted in dealing with fault detection issues, both in the research and practical application domains. The mostly convincing argument for their popular use is their simple computation form that is parameterised by the mean and covariance matrix of the measurement variables under consideration. In other words, once the mean and covariance matrix are determined (or estimated), the computation of χ^2- or T^2-test statistics is straightforward. From the statistical point of view, χ^2- and T^2-test statistics are the Mahalanobis distance that can be used as a dissimilarity measure between two random vectors of the same distribution with the identical covariance matrix. Initially, the Mahalanobis distance is used for checking the deviation of a measurement point (data) from the given mean (center)

[1]. Applying it for fault detection, the associated test statistics can be interpreted as the dissimilarity measure between the fault-free and the faulty operations.

Our work in this chapter on the alternatives to χ^2- and T^2-test statistics is initially an answer to the question arising from the use of χ^2- and T^2-test statistics: how to deal with the case that the mean and covariance matrix are varying and corrupted with uncertainties. Our ambition for an optimal fault detection drives us, on the other hand, to present a general (alternative) solution for optimally detecting faults which may cause changes in any parameter of a distribution rather than in mean only, for which χ^2- and T^2-test statistics knowingly provide the best solution.

Our work consists of two parts. The first part is dedicated to the general solution for optimal fault detection. Motivated by the Neyman-Pearson Lemma, we propose to apply the GLR method and maximal likelihood ratio (MLR) for building the test statistic, which would lead to the optimal fault detection performance in the sense of maximising the fault detectability when false alarm rate is limited to an acceptable level. Two major problems we have to face by the implementation of the GLR algorithm are

- MLE of the parameters in the faulty distribution (using the online measurement data) and
- the threshold setting.

The first one is an optimisation problem for which extensive results can be found in the literature. In case of normal distribution the MLE of the mean (vector) and covariance matrix are well-known, as demonstrated in our example study. The second problem is the difficulty with the probability computation based on the MLR, which is necessary for the determination of the threshold, even if the PDFs of the fault-free and faulty distributions are known. As a solution, we propose to use the so-called RA-technique to determine the threshold. The RA-technique and its application to fault detection and diagnosis including threshold settings are the topics to be addressed in the next part (Chaps. 16–18). For instance, Algorithm 18.1 presented in Chap. 18 can be applied for the purpose of threshold setting.

The Kullback-Leibler divergence is a well-known dissimilarity measure between two distributions [2]. Inspired by the idea of formulating fault detection problems as checking the dissimilarity between two distributions, KL divergence has been adopted as a test statistic and applied in the fault detection and diagnosis research recently [3–6]. In our work, we have given an empirical algorithm for the application of KL divergence for fault detection, in which the threshold setting can be realised in a data-driven fashion or by using the RA-technique. It is remarkable that we have proposed to adopt the KL divergence from the fault-free distribution to the faulty distribution as the test statistic, different from the existing works. Due to the asymmetry of KL divergence, it is known that the KL divergence from fault-free distribution to the faulty distribution is generally different from the one from the faulty distribution to the fault-free distribution. Motivated by this observation, the relationships between the GLR and KL divergence based test statistics and fault detection schemes are investigated. This work is performed along the line in the study by Eguchi and Copas [7]. Eguchi and Copas have demonstrated that

- KL divergence is indeed the expectation of the GLR and
- the empirical realisations of the KL divergence and the GLR are, up to a constant, equivalent.

Under consideration that the GLR based fault detection scheme results in, according to Neyman-Pearson Lemma, optimal fault detectability, the results of this study reveal that the KL divergence from the fault-free distribution to the faulty distribution is a more reasonable and convincing test statistic.

The last topic in the first part of our study is the asymptotic behaviour of the KL divergence and the GLR based test statistics. It is the common results in statistics [8, 9] that

- the MLE of parameters converges (in probability) to their true value of the distribution, from which the samples are generated, and
- the estimation error multiplied by $\sqrt{n}$ converges to zero-mean normal distribution with the inverse of the fisher information matrix [10] as covariance, where n is the number of the i. i. d. samples.

Based on these results, it can be proved that

- KL divergence and MLR as test statistics can be well approximated, by sufficient large number of samples, by χ^2 distribution with the degrees of freedom equal to the number of the (estimated) parameters, and
- this approximation is of high accuracy in case of incipient faults.

It is verified again that the KL divergence from the fault-free distribution to the faulty distribution should be adopted for the fault detection purpose.

The Hoeffding's inequality given in Lemma 15.1 is a well-known result in statistics, see, for instance, [11].

The second part of our study consists in the effort to find data evaluation functions and the associated fault detection schemes aiming at efficiently dealing with possible variations or uncertainties in the process variables under consideration. Also, no assumption on the statistical distributions of the process variables should be made in this investigation. That is, we are seeking for a data evaluation function as an alternative to the existing test statistics. Moreover, the data should be presented in a format so that there is no information loss. As a proper data format we have decided to adopt the empirical second moment of the measurement data given by

$$Y = \left[y(1) \cdots y(l) \right] \in \mathcal{R}^{m \times l}, P = \frac{1}{l} Y Y^T \in \mathcal{R}^{m \times m}, l >> m.$$

It is worth mentioning that industrial data are often collected batchwise. Hence, matrix P is a natural format without involved data pre-processing. Note that different from the existing test statistics and the associated fault detection schemes, in which normal process models (distributions) and variations are handled separately, matrix P includes all information about normal operations, faults, unexpected variations and uncertainties. It is our intention to distinguish normal operations with uncertainties and faulty operations in terms of changes in P matrix. To this end, a powerful (mathematical) tool is needed. On the assumption that P is positive-definite, the collected data in the SPD format form a $\frac{m(m+1)}{2}$ dimensional manifold $\mathcal{P}(m)$. This allows us to apply the existing differential-geometric methods, rather than MVA, as a tool for our problem solutions. For our purpose, we have first introduced very basic differential-geometric properties of $\mathcal{P}(m)$ as a Riemannian manifold as well as some relevant concepts and methods in Sub-section 15.4.1. The reader is referred to [12, 13] for essential knowledge of differential-geometric methods and Riemannian manifolds. All definitions and relations of the tangent space, geodesic curves, exponential and logarithmic maps as well as Riemannian distance in $\mathcal{P}(m)$ can be found in [14–16]. The definition (15.37) in Remark 15.3 is given in [17]. The definition of geometric mean and the existence condition given in Theorem 15.1 play a central role in developing fault detection schemes. We refer the reader to [14, 15] for the detailed description and the associated algorithm for the computation of the geometric mean. The proof of Theorem 15.1 can be found in [14].

In two steps, the mathematical results on Riemannian manifold $\mathcal{P}(m)$ are applied to dealing with fault detection issues. In the first step, a basic fault detection scheme is proposed in Sub-section 15.4.2, whose core is the use of the Riemannian distance between the geometric mean of the (collected) data sets and a point (data set) in $\mathcal{P}(m)$ as the evaluation function. A modified version of this algorithm by concentrating the data is given in Algorithm 15.5. Following the idea proposed by Fletcher [18] to model the data points (sets) in a Riemannian manifold using geodesic regression, the basic fault detection scheme is extended, in which the tangent vector is multiplied by a measurement variable. In this way, we are able to model variations caused by the changes in the operations in terms of the tangent vector together with the (scalar) variable representing the operation condition. This allows us to build an evaluation function as a function of the operation conditions. For the realisation of the relevant computations, the reader is, for instance, referred to [19].

In order to deal with typical multimode operations of industrial processes, we have studied, in the second step, clustering on Riemannian manifolds and its application to fault detection and diagnosis. To this end, the standard k-means clustering algorithm [20] is applied with the modification that the Euclidean distance is substituted by the Riemannian distance, and presented as Algorithm 15.7. Considering that some clusters may be sparse with their support points, the convex model recently proposed by Zhao et al. [21] is adopted. The convex model can be interpreted as multivariate interpolations based on the support points and thus used for modelling the operations represented by the clusters. In order to gain insight into the interpolation on the Riemannian manifold $\mathcal{P}(m)$ and its re-formulation as a weighted mean problem, we have studied a special case with two support points in a cluster. Lemma 15.2 is a result given in [14] and the result in Lemma 15.3 is a part of the proof of Theorem 15.1 given by [14] as well. At the end of our work, we have introduced the definitions of the distance from a point to a sub-manifold on the Riemannian manifold $\mathcal{P}(m)$ as well as the distance between two sub-manifolds on the Riemannian manifold $\mathcal{P}(m)$. These two definitions as well as the iterative algorithm for the computation of the distance between two sub-manifolds are inspired by the definitions related to the divergences from a point or from a sub-manifold to a sub-manifold as well as their computations introduced by Amari in [13]. These definitions and their computations are essential for clustering based fault detection and diagnosis.

At the end of this chapter, we have presented three examples. While the first two examples serve for illustrating the theoretical results and fault detection schemes in the regard of the Riemannian manifold $\mathcal{P}(m)$, the third example deals with an application of the proposed Riemannian distance based fault detection schemes to achieving performance monitoring in control systems, both in the data-driven and model-based fashions. To our best knowledge, no research result has been reported on this topic. It can be expected that sophisticated investigations on Riemannian distance based fault detection would make valuable contributions to the fault diagnosis technique for dynamic systems.

References

[1] R. D. Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, pp. 1–18, 2000.

[2] S. Kullback, *Information Theory and Statistics*. John Wiley and Sons, 1959.

[3] J. Zeng, U. Kruger, J. Geluk, X. Wang, and L. Xie, "Detecting abnormal situations using the Kullback-Leibler divergence," *Automatica*, vol. 50, pp. 2777–2786, 2014.

[4] J. Marmouche, C. Delpha, and D. Diallo, "Incipient fault detection and diagnosis based on kullback-leiber divergence using principal component analysis: Part I," *Signal processing*, vol. 94, pp. 278–287, 2014.

[5] L. Xie, J. Zeng, U. Kruger, X. Wang, and J. Geluk, "Fault detection in dynamic systems using the kullback-leiber divergence," *Control Engineering Practice*, vol. 43, pp. 39–48, 2015.

[6] A. Youssef, C. Delpha, and D. Diallo, "An optimal fault detection threshold for early detection using Kullback-Leibler divergence for unknown distribution data," *Signal Processing*, vol. 120, pp. 266–279, 2016.

[7] S. Eguchi and J. Copas, "Interpreting Kullback-Leibler divergence with the neyman-pearson lemma," *Journal of Multivariate Analysis*, vol. 97, pp. 2034–2040, 2006.

[8] J. Pfanzagl, *Parametric Statistical Theory*. Walter de Gruyter, 1994.

[9] W. Newey and D. McFadden, *Large Sample Estimation and Hypothesis Testing, in Handbook of Econometrics*. Elsevier Science, 1994, ch. 36, pp. 2111–2245.

[10] E. Lehmann and G. Casella, *Theory of Point Estimation (2nd Ed.)*. Springer, 1998.

[11] R. Tempo, G. Calafiro, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems, Second Edition*. London: Springer, 2013.

[12] W. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry (2nd Ed.)*. London: Avademic Press, 1986.

[13] S. Amari, *Information Geometry and its Applications*. Japan: Springer, 2016.

[14] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, pp. 735–747, 2005.

[15] M. Moakher and P. Batchelor, *Symmetric Positive-Definite Matrices: From Geometry to Applications and Visualization*, 2006, ch. 17 in Book Visualization and Processing of Tensor Fields, pp. 285–298.

[16] J.-B. Hiriart-Urruty and J. Malick, "A fresh variational-analysis look at the positive semidefinite matrices world," *J. Optim. Theory Appl.*, vol. 153, pp. 551–577, 2012.

[17] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadephia: SIAM, 2008.

[18] P. T. Fletcher, "Geodesic regression and the theory of least squares on riemannian manifolds," *Int. Journal of Computer Vision*, vol. 105, pp. 171–185, 2013.

[19] Q. Rentmeesters, "A gradient method for geodesic data fitting on some symmetric riemannian manifolds," *the 50th IEEE CDC*, pp. 7141–7146, 2011.

[20] J. Hartigan and M. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Jounal of the Royal Statistical Society, Series C*, vol. 28, pp. 100–108, 1979.

[21] K. Zhao, A. Wiliem, S. Chen, and B. Lovell, "Convex class model on symmetric positive definite manifolds," *arXiv:1806.05343v2*, 2019.

Part V
Application of Randomised Algorithms to Assessment and Design of Fault Diagnosis Systems

Chapter 16
Probabilistic Models and Randomised Algorithms

16.1 Motivation

In the previous chapters, we have introduced numerous fault detection and diagnosis methods. In fact, there are a huge number of fault diagnosis methods published in the recent decade. This rapid development and the amazingly increasing number of publications provide us with rich theoretical solutions for most fault diagnosis issues. On the other hand, engineers and researchers may have the similar experience that it is often a hard work to find a right one among a great number of available fault diagnosis methods, when we are facing a practical fault detection case. The following two reasons, among numerous possible ones, may call our attention:

- only few approaches are dedicated to the design of fault diagnosis systems on practical demands for such systems,
- although most of existing design approaches should contain certain novelty in designing a fault diagnosis system, a direct comparison of these or some of these approaches seems difficult.

In industrial applications, performance of a fault diagnosis system is generally measured/quantified by false alarm rate (FAR), fault detection rate (FDR) or mean time to fault detection (MT2FD), as suggested by industrial recommendations, regulations and guidelines. This aspect has been described in Chap. 2, and also in many monographs on fault diagnosis. Unfortunately, only few research efforts to integrate such performance criteria into the fault diagnosis system design have been reported, in particular in case of dealing with model- and observer-based fault diagnosis design methods.

Benchmark (case) study is a popular way of comparing different fault diagnosis methods. For instance, Tennessee Eastman Process (TEP) is a mostly used benchmark process for comparison studies on data-driven fault diagnosis methods. Due to the limitation of simulation capacity and data amount, in most TEP-based benchmark studies the FAR and FDR computations have been realised on certain assumptions

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_16

and based on approximation. In general, results from a benchmark study are, although useful and valuable, less representative in the statistical sense, often due to the technical specifications of the benchmark process under consideration.

False alarms are caused by uncertainties like disturbances, parameter variations etc. within and around the process under supervision, while fault detectability strongly depends on the size, form or the energy level of the faults to be detected. Viewing from this aspect, it is reasonable to study FAR, FDR and MT2FD issues in the probabilistic framework, since both uncertainties and faults are in their nature random variables.

Motivated by the above observations and considerations, in this part we try to establish a probabilistic framework to deal with assessment and design issues of fault diagnosis systems. This framework will consist of three functional levels:

I. probabilistic models for faults and uncertainties, which will be introduced in this chapter,
II. performance assessment of fault diagnosis systems in terms of FAR, FDR and MT2FD as well as the associated computation algorithms. These issues will be addressed in Chap. 17, and
III. design of fault diagnosis systems in the context of trade-off between FAR and FDR. This topic will be handled in the last chapter of this part.

The fundament for our work is probabilistic methods, which have been successfully applied to the analysis and design of robust control systems. With the help of the technique for generating random samples, control performance of uncertain systems can be evaluated in the context of probability or expectation value which are estimated using the generated random samples. Such algorithms are called randomised algorithms (RA). They will also be adopted in our work and introduced at the end of this chapter.

16.2 Probabilistic Models for Uncertain Systems

The objective of introducing probabilistic models is to deal with uncertainties in system models in a probabilistic framework. In order to model different types of uncertainties, including parameter variations, disturbances as well as their combinations, we introduce three different model forms. They are

- probabilistic parameter model (PPM),
- probabilistic time function parameter model (PTFPM) and
- probabilistic uncertainty mode model (PUMM).

16.2.1 Probabilistic Parameter Model

A PPM describes the random (uncertain) behaviour of the parameters of a process model under consideration. Let θ_i be a parameter vector or matrix of a process model and write it as

$$\theta_i = \theta_{io} + \Delta P_i \Delta\theta_i, \tag{16.1}$$

where θ_{io} is known and constant, represents the nominal operation value and is the mean of θ_i, $\Delta\theta_i \in \left[\Delta\theta_{i,-}, \Delta\theta_{i,+}\right]$ is a random variable (vector or matrix) representing possible variations of the parameter around its mean. It is assumed that the distribution of $\Delta\theta_i$ is known. ΔP_i is a known constant matrix. As a result, the distribution of θ_i is known as well.

Example 16.1 *The probabilistic model adopted in the PPCA method, which is introduced in Sub-section 3.5.4,*

$$\begin{aligned} &y = Ex + \varepsilon \in \mathcal{R}^m, x \in \mathcal{R}^n, rank\,(E) = n, m > n, \\ &\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I), x \sim \mathcal{N}(0, I), \end{aligned}$$

is an example of the PPM in (16.1). In the above model, Ex can be viewed as the (vector-valued) mean of y, which has a nominal value $\theta_o = 0$, and random variation is described by

$$\Delta P \Delta\theta = Ex \sim \mathcal{N}(0, EE^T).$$

Example 16.2 *Another example is the so-called polytopic type uncertainty widely used in the robust control research. Consider the process model*

$$x(k+1) = Ax(k) + Bu(k), y(k) = Cx(k) + Du(k), \tag{16.2}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix} + \sum_{i=1}^{\xi} \Delta\theta_i \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}.$$

The system matrices A, B, C, D are of appropriate dimensions with A_o, B_o, C_o, D_o denoting the known nominal system matrices. The matrix

$$\Delta = \sum_{i=1}^{\xi} \Delta\theta_i \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$$

represents the polytopic uncertainty, which can be viewed as a special case of the PPM with

$$\theta_o = \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix}, \Delta P_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}.$$

16.2.2 Probabilistic Time Function Parameter Model

Uncertainties could be caused by manufacturing error of a system component or by linearisation error of a nonlinear function around an operating point, or by disturbances and noises. In case that they are time functions, we introduce the PTFPM to describe the random behaviour in the parameters of such time functions. In our work, we only consider l_2-bounded functions. Let

$$\phi_i(k, \theta_i) \in \mathcal{H}_2, i \in \{1, 2, \cdots\}. \tag{16.3}$$

θ_i is the parameter (vector) of the time function. It is assumed that θ_i is a random variable whose distribution is known and $\theta_i \in [\theta_{i,-}, \theta_{i,+}]$.

Example 16.3 *The most popular forms of such time functions are*

$$\begin{gathered}
\textit{off-set function: } \phi_i(k, \theta_i) = \theta_i \sigma_T(k), \\
\textit{ramp function: } \phi_j(k, \theta_j) = \theta_j k \sigma_T(k), \\
\textit{exponential function: } \phi_l(k, \theta_l) = \theta_{l,1} e^{\theta_{l,2} k} \sigma_T(k), \theta_l = \begin{bmatrix} \theta_{l,1} \\ \theta_{l,2} \end{bmatrix}, \\
\sigma_T(k) = \begin{cases} 1, 0 \leq k \leq T, \\ 0, \textit{ otherwise,} \end{cases}
\end{gathered}$$

or their combinations.

We denote the set of all parameters of the probabilistic parameter models, PPM and PTFPM, for uncertainties by Θ.

A key step in a randomised algorithm is the generation of random samples according to a given distribution. It is known that distributions of the random variables may remarkably influence the final result returned by the applied RA. In order to establish a common basis for a fair performance assessment, it is necessary to find a "most random" distribution. In information theory, this is the maximum entropy problem. It is well known that the uniform distribution on the interval $[a, b]$ is the maximum entropy distribution among all distributions which are supported in the interval $[a, b]$. Also, studies have demonstrated that uniform distribution can be applied in the robustness analysis if, in the worst-case, no distribution knowledge is available. In considering these arguments, we assume that all random parameters defined in PPM and PTFPM are uniformly distributed. It should be pointed out that any other distributions can be used for generating random samples when they are known.

16.2.3 Probabilistic Uncertainty Mode Model

Motivated by the fact that different types of model uncertainties and unknown inputs may be present in the system in different combinations, we introduce M uncertainty modes $\Theta_i, i = 1, \cdots, M$,

$$\begin{aligned}\Theta_i &= \{\theta_i \text{ in PPM (16.1)}, i = 1, \cdots, l\} \cup \{\theta_i \text{ in PTFPM (16.2)}, i = 1, \cdots, \kappa\} \\ &=: \left\{\theta_j^{(i)} \in \Theta, j = 1, \cdots, \alpha_i\right\} \end{aligned} \tag{16.4}$$

representing different combinations of the model uncertainties and unknown inputs, and call them PUMM. In our subsequent study, we denote the support and PDF of $\theta_j^{(i)}$ by $\mathcal{D}_{\theta_j^{(i)}}$ and $D_{i,j}$, respectively.

We assume that process knowledge is available about the present frequency of each uncertainty mode during process operation and express it in terms of a (discrete) random variable Θ with M values, $\Theta_1, \cdots, \Theta_M$, and its probabilistic mass function (PMF),

$$\Pr\left(\Theta = \Theta_i\right), i = 1, \cdots, M, \tag{16.5}$$

is thus known. Note that for $M = 1$, we have

$$\Theta = \Theta_1, \Pr\left(\Theta = \Theta_1\right) = 1.$$

This is the situation that all possible model uncertainties and unknown inputs are present in the process simultaneously, a popular way of today's handling of uncertainties and disturbances.

16.3 Probabilistic Fault and Evaluation Function Models

16.3.1 Probabilistic Fault Models

In practice, faults are present in a process either in additive or in multiplicative form. In general, the emergence of a fault is a dynamic process. The emergence rate, the size of the fault or its form and direction (distribution) could be random, depending on the operation conditions and relevant process components. Under this consideration, we model each fault, analogue to the PTFPM, by

$$f_i\left(k, \theta_{f_i}\right) \in \mathcal{H}_2, i \in \{1, \cdots, \}. \tag{16.6}$$

Here, θ_{f_i} is the parameter (vector) of the time function, which is a random variable with a known distribution. The most reasonable form for such a time function is

$$f_i\left(k,\theta_{f_i}\right)=\theta_{f_i,1}\left(1-e^{\theta_{f_i,2}k}\right)\sigma_T(k),\theta_{f_i}=\begin{bmatrix}\theta_{f_i,1}\\ \theta_{f_i,2}\end{bmatrix},$$

where $\theta_{f_i,1}$ indicates the size (and direction, when $f_i\left(k,\theta_{f_i}\right)$ is vector-valued), while $\theta_{f_i,2}$ determines the emergence rate. We denote the set of all parameters of the probabilistic fault model (16.6) by Θ_f.

It is natural that faults may be present in a process in different fault patterns which can be modelled as combinations of the probabilistic fault functions given in (16.6). We suppose, there are K fault patterns, $\Theta_{f,i}, i=1,\cdots,K$, representing possible simultaneous combinations

$$\begin{aligned}\Theta_{f,i}&=\left\{\theta_{f_j},\theta_{f_j}\text{ in model (16.6)}, j\in\{1,\cdots,\}\right\}\\&=:\left\{\theta_{f_j}^{(i)}\in\Theta_f, j=1,\cdots,\beta_i\right\},\end{aligned}\tag{16.7}$$

and the support and PDF of $\theta_{f_j}^{(i)}$ are denoted by $\mathcal{D}_{\theta_{f_j}^{(i)}}$ and $D_{f_j,i}$, respectively.

For our purpose of addressing different fault patterns, we assume $\Theta_f=\left\{\Theta_{f,1},\cdots,\Theta_{f,K}\right\}$ is a (discrete) random variable, and the probability of a fault in pattern $\Theta_{f,i}, i=1,\cdots,K$, is known and denoted by

$$\Pr\left(\Theta_f=\Theta_{f,i}\right), i=1,\cdots,K.$$

It is worth remarking that the determination of $\Pr\left(\Theta_f=\Theta_{f,i}\right)$ requires *a prior* knowledge of e.g. failure rates of components embedded in the system like sensors, actuators etc. In fact, an intelligent maintenance system would deliver such information. In case that no knowledge is available, it can be, according to the maximum entropy principle, assumed that

$$\Pr\left(\Theta_f=\Theta_{f,i}\right)=1/K.$$

To support our subsequent study, the major notations introduced above are summarised in the following table (Table 16.1).

Table 16.1 Notations adopted in the probabilistic models

Θ, Θ_f	Parameter sets of the probabilistic models
$\Theta_i, \Theta_{f,i}$	Uncertainty mode and fault pattern
$\theta_j^{(i)}, \theta_{f_j}^{(i)}$	The j-th parameter in $\Theta_i, \Theta_{f,i}$
α_i, β_i	No. of the parameters in $\Theta_i, \Theta_{f,i}$
$\mathcal{D}_{\theta_j^{(i)}}, \mathcal{D}_{\theta_{f_j}^{(i)}}$	Supports of $\theta_j^{(i)}$ and $\theta_{f_j}^{(i)}$
$D_{i,j}, D_{f_j,i}$	PDFs of $\theta_j^{(i)}$ and $\theta_{f_j}^{(i)}$

16.3.2 Probabilistic Models of Test Statistics and Evaluation Functions

Let J denote a test statistic or a residual evaluation function. It is clear that J is also a function of the uncertainties and faults in the process under monitoring. It follows from the probabilistic models of systems with uncertainties and faults that J can also be modelled as

$$J = \mathcal{J}(\Theta) \tag{16.8}$$

during fault-free operations and

$$J = \mathcal{J}\left(\Theta, \Theta_f\right) \tag{16.9}$$

in the faulty case. As a function of random variables Θ, Θ_f, J is a random variable as well. In fact, the definitions and computations of FAR, FDR and MDR will be introduced based on the models (16.8) and (16.9).

16.4 Preliminaries of Randomised Algorithms

Recalling the definitions of FAR, FDR and MDR given in Chap. 2 makes it clear that the core of their computations consists in the determination of the probability that $J > J_{th}$ under certain (fault) conditions. This problem is similar to the RA-based probabilistic performance verification studied in robust control, see for instance the references given at the end of this chapter. Schematically, the problem to be addressed can be formulated in two different forms: Given a performance function $J(x)$, random variable x with the known density $D(x)$ and support $\mathcal{D}_x$,

- finding an estimate $\hat{p}(\gamma)$ for the probability

  $$p(\gamma) = \Pr\left(J(x) \le \gamma\right),$$

  which should satisfy

  $$\left|p(\gamma) - \hat{p}(\gamma)\right| < \epsilon \text{ or } p(\gamma) < \hat{p}(\gamma) + \epsilon \tag{16.10}$$

  with probability at least $1 - \delta$,
- finding an estimate $\hat{\gamma}_{\max}$ of the maximum value of $J(x)$, satisfying, with probability greater than $1 - \delta$,

$$p(\hat{\gamma}_{\max}) = \Pr\left(J(x) \le \hat{\gamma}_{\max}\right) \ge 1 - \epsilon, \tag{16.11}$$

where $\epsilon \in (0, 1)$ is the given accuracy requirement and $1 - \delta$ the confidence level with $\delta \in (0, 1)$.

In the RA framework, the estimate $\hat{p}(\gamma)$ will be computed using N random samples of x. Below are some elemental results for the RA-based estimation of $p(\gamma)$ and the determination of the needed sample number N given in the monograph by Tempo et al.

Given random variable x with known PDF $D(x)$ and support $\mathcal{D}_x$, it holds

$$p(\gamma) = \Pr\left(J(x) \le \gamma\right) = \int\limits_{\mathcal{D}_\gamma} D(x)dx,$$
$$\mathcal{D}_\gamma = \{x \in \mathcal{D}_x : J(x) \le \gamma\} \subset \mathcal{D}_x.$$

For the estimation of $p(\gamma)$, N i.i.d. random samples, $x^{(1)}, \cdots, x^{(N)} \subset \mathcal{D}_x$, are first generated, and correspondingly $J\left(x^{(i)}\right), i = 1, \cdots N$, are calculated. An estimate for $p(\gamma)$ is then given by

$$\hat{p}(\gamma) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{\mathcal{D}_\gamma}\left(x^{(i)}\right), \mathbb{I}_{\mathcal{D}_\gamma}\left(x^{(i)}\right) = \begin{cases} 1, & \text{if } x^{(i)} \in \mathcal{D}_\gamma, \\ 0, & \text{otherwise.} \end{cases} \tag{16.12}$$

It is evident that the accuracy and reliability of estimate $\hat{p}(\gamma)$ depend on the sample number N. In the following theorems, some results on the determination of N are summarised.

Theorem 16.1 *(Hoeffding's inequality) Let $x_i \in [a_i, b_i], i = 1, \cdots, N$, be independent bounded random variables. For any $\epsilon > 0$, it holds*

$$\Pr\left(\sum_{i=1}^{N} x_i - \mathcal{E}\left(\sum_{i=1}^{N} x_i\right) \ge \epsilon\right) \le e^{-\frac{2\epsilon^2}{\sum_{i=1}^{N}(b_i - a_i)^2}}, \tag{16.13}$$

$$\Pr\left(\sum_{i=1}^{N} x_i - \mathcal{E}\left(\sum_{i=1}^{N} x_i\right) \le -\epsilon\right) \le e^{-\frac{2\epsilon^2}{\sum_{i=1}^{N}(b_i - a_i)^2}}. \tag{16.14}$$

It is straightforward that, for $[a_i, b_i] = [0, 1]$, we have

- two-sided Chernoff bound

$$N \ge \frac{1}{2\epsilon^2}\log\frac{2}{\delta} \Longrightarrow \Pr\left(\left|p(\gamma) - \hat{p}(\gamma)\right| < \epsilon\right) > 1 - \delta, \tag{16.15}$$

- one-sided Chernoff bound

$$N \ge \frac{1}{2\epsilon^2}\log\frac{1}{\delta} \Longrightarrow \Pr\left(p(\gamma) < \hat{p}(\gamma) + \epsilon\right) > 1 - \delta, \tag{16.16}$$

where $\hat{p}(\gamma)$ is the random variable defined in (16.12).

Theorem 16.2 *For any* $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, *if*

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}}, \tag{16.17}$$

then, with probability greater than $1 - \delta$, *we have*

$$\Pr\left(J\left(x\right) \leq \hat{\gamma}_N\right) \geq 1 - \epsilon, \hat{\gamma}_N = \max_{i=1,\cdots,N} J\left(x^{(i)}\right), \tag{16.18}$$

where $J\left(x\right)$ *is a cost function with random variable* x, *and* $x^{(i)}, i = 1, \cdots, N$, *are* N *i.i.d. samples of* x *generated according to its PDF,* $D(x)$, *with support* $\mathcal{D}_x$.

16.5 Notes and References

The objective of this part is to establish a probabilistic framework for the performance analysis, assessment and design of fault detection systems. Although in many industrial sectors like aerospace, automotive and process industries it is recommended to assess fault diagnosis performance in the probabilistic context [1–4], few attention is paid to this issue in the research domain, in particular in the model-based thematic field.

Today's common way to demonstrate the application performance of a fault detection method or system is to perform a benchmark (case) study and to use the achieved results as an assessment of the fault diagnosis performance. Tennessee Eastman Process [5, 6] is a mostly used benchmark process for such studies. On the other hand, it is clear that such benchmark studies are less representative in the statistical sense, even though the TE benchmark platform is well-established.

As the first step towards the probabilistic framework, we have introduced, in this chapter, diverse probabilistic models for uncertainties and faults. They build the basis level of the probabilistic framework and allow us to have a fair assessment of fault diagnosis performance using the concepts like FAR, FDR and MT2FD. Note that the probabilistic models presented in this work can be divided into two levels: (i) the functional and parameter (lower) level, and (ii) the mode and pattern (higher) level.

In the second part of this chapter, we have briefly introduced randomised algorithms, a probabilistic method, which has been successfully applied to the analysis and design of robust control systems [7–9]. The basic idea behind this application is the randomisation of uncertainties in the control system under consideration and, based on it, the estimation of system performance. Analogue to it, we will apply the RA-method to the computation of FAR, FDR and MT2FD and further to the design of fault detection systems.

The brief introduction to the essentials of the RA-method in Sect. 16.4 can be found in the monograph by Tempo et al. [9]. Theorems 16.1–16.2 are well-known and can be found, for instance, in [9] as well. We refer the reader to [7–9] for more details and advanced study on this topic.

References

1. SAE, "SAE AIR4985: A methodology for quantifying the performance of an engine monitoring system," *SAE International*, 2012.
2. G. Box, S. Graves, S. Bisgaard, J. V. Gilder, K. Marko, J. James, M. Seifer, M. Poublon, and F. Fodale, "Detecting malfunctions in dynamic systems," *Transactions of the society of automotive engineerings, SAE Technical paper series*, vol. 2000, pp. 1–11, 2000.
3. K. Reif and K. H. Dietsche, *Kraftfahrttechnisches Tschenbuch (in German)*. Vieweg + Teubner, 2014.
4. NAMUR, "Alarm management," *NAMUR Recommendations*, vol. NA 102, 2003.
5. N. Ricker, "Decentralized control of the tennessee eastman challenge process," *Journal of Process Control*, vol. 6, pp. 205–221, 1996.
6. S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, pp. 1567–1581, 2012.
7. M. Vidyasagar, "Randomized algorithms for robust controller synthesis using statistical learning theory," *Automatica*, vol. 37, pp. 1515–1528, 2001.
8. G. Calafiore, F. Dabbene, and R. Tempo, "Research on probabilistic methods for control system design," *Automatica*, vol. 47, pp. 1279–1293, 2011.
9. R. Tempo, G. Calafiro, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems, Second Edition*. London: Springer, 2013.

Chapter 17
Assessment of Fault Detection Performance and Computation Algorithms

In Chap. 2, definitions of the important indicators for fault detection performance like FAR and FDR have been introduced. They will be re-viewed in this chapter in the context of the probabilistic models presented in the last chapter. On this basis, RA-aided computation algorithms will be investigated. Moreover, we will also study the definition and computation of MT2FD.

17.1 False Alarm Rate and RA-aided Assessment

A false alarm is triggered by uncertainties in the process under monitoring. Considering the probabilistic model (16.8) for test statistics and residual evaluation functions, the condition for a false alarm is

$$\mathcal{J}(\Theta) > J_{th}, \Theta_f = O,$$

under the use of the detection logic

$$\begin{cases} \mathcal{J}(\Theta) > J_{th} \Longrightarrow \text{faulty}, \\ \mathcal{J}(\Theta) \leq J_{th} \Longrightarrow \text{fault-free}. \end{cases} \tag{17.1}$$

Here, $\Theta_f = O$ is the notation for fault-free operations. On the other hand, according to the probabilistic models, the uncertainties can be present in the process in different modes. Under this consideration, we introduce below two definitions for FAR.

Definition 17.1 *Given a fault detection system with evaluation function (or test statistic) $\mathcal{J}(\Theta)$, threshold J_{th} and detection logic (17.1), we call the conditional probability*

$$p_{FAR}(\Theta_i) := \Pr(\mathcal{J}(\Theta) > J_{th} | \Theta = \Theta_i)$$

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_17

false alarm rate (FAR) with respect to (w.r.t.) uncertainty mode Θ_i, and the marginal probability

$$p_{FAR}(\Theta) = \sum_{i=1}^{M} p_{FAR}(\Theta_i) \Pr\left(\Theta = \Theta_i\right)$$

average false alarm rate (AFAR).

Next, we study the FAR computation issues. Note that the support and PDF of Θ_i, $\mathcal{D}_{\Theta_i}$ and D_{Θ_i}, are

$$\mathcal{D}_{\Theta_i} = \mathcal{D}_{\theta_1^{(i)}} \cup \cdots \cup \mathcal{D}_{\theta_{\alpha_i}^{(i)}}, \; D_{\Theta_i}(\theta) = \prod_{j=1}^{\alpha_i} D_{i,j}(\theta).$$

For our purpose, let

$$\mathcal{D}_{FA}(\Theta_i) = \left\{\theta, \theta \in \mathcal{D}_{\Theta_i}, \mathcal{J}(\Theta_i) > J_{th}\right\} \subset \mathcal{D}_{\Theta_i}$$

be the parameter set of those uncertainties which trigger false alarms. As a result, it holds

$$p_{FAR}(\Theta_i) = \int\limits_{\mathcal{D}_{FA}(\Theta_i)} D_{\Theta_i}(\theta) d\theta. \tag{17.2}$$

Now, applying the RA-based estimation of a probability given in Sub-section 16.4, the FAR w.r.t. uncertainty mode Θ_i can be estimated as follows

$$\hat{p}_{FAR}(\Theta_i) = \frac{1}{N} \sum_{j=1}^{N} \mathbb{I}_{\mathcal{D}_{FA}}\left(\theta_j^{(i)} \,|\Theta_i\right), \tag{17.3}$$

$$\mathbb{I}_{\mathcal{D}_{FA}}\left(\theta_j^{(i)} \,|\Theta_i\right) = \begin{cases} 1, \text{ if } \theta_j^{(i)} \in \mathcal{D}_{FA}(\Theta_i), \\ 0, \text{ otherwise}, \end{cases} \tag{17.4}$$

where $\theta_j^{(i)}$, $j = 1, \cdots, N$, are i.i.d. random samples generated from $\mathcal{D}_{\Theta_i}$ according to the known PDF $D_{\Theta_i}(\theta)$. It follows from the two-sided Chernoff bound (16.15) given in Theorem 16.1, the number of the samples is set to be

$$N \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta},$$

for given ϵ, δ, to ensure that

$$\Pr\left(\left|\hat{p}_{FAR}(\Theta_i) - p_{FAR}(\Theta_i)\right| < \epsilon\right) > 1 - \delta. \tag{17.5}$$

The following algorithm is the realisation of $\hat{p}_{FAR}(\Theta_i)$ given in (17.3).

Algorithm 17.1 *FAR estimation w.r.t. uncertainty mode Θ_i: Given $\epsilon \in (0, 1), \delta \in (0, 1)$,*

```
• Set integer $N \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$;
• Generate N samples $\theta_j^{(i)}$, $j = 1, \cdots, N$, according to $D_{\Theta_i}(\theta)$;
• Set $n = 0$;
• For $j = 1$ to N
    – Compute $\mathcal{J}\left(\theta_j^{(i)}\right)$ (by means of simulation)
    – If
          $\mathcal{J}\left(\theta_j^{(i)}\right) > J_{th}$,
      set
          $n = n + 1$;
• End for
• Return
          $\hat{p}_{FAR}(\Theta_i) = \frac{n}{N}$.
```

Note that setting

$$\hat{p}_{FAR}(\Theta) = \sum_{i=1}^{M} \hat{p}_{FAR}(\Theta_i) \Pr(\Theta = \Theta_i)$$

yields

$$\begin{gathered}
\hat{p}_{FAR}(\Theta) - p_{FAR}(\Theta) = \sum_{i=1}^{M} \left(\hat{p}_{FAR}(\Theta_i) - p_{FAR}(\Theta_i)\right) \Pr(\Theta = \Theta_i) \Longrightarrow \\
\left|\hat{p}_{FAR}(\Theta) - p_{FAR}(\Theta)\right| \leq \sum_{i=1}^{M} \left|\hat{p}_{FAR}(\Theta_i) - p_{FAR}(\Theta_i)\right| \Pr(\Theta = \Theta_i).
\end{gathered}$$

Hence,

$$\begin{gathered}
\left|\hat{p}_{FAR}(\Theta_i) - p_{FAR}(\Theta_i)\right| < \epsilon \Longrightarrow \\
\sum_{i=1}^{M} \left|\hat{p}_{FAR}(\Theta_i) - p_{FAR}(\Theta_i)\right| \Pr(\Theta = \Theta_i) < \epsilon \sum_{i=1}^{M} \Pr(\Theta = \Theta_i) = \epsilon \Longrightarrow \\
\left|\hat{p}_{FAR}(\Theta) - p_{FAR}(\Theta)\right| < \epsilon,
\end{gathered}$$

which results in

$$\begin{aligned}
\forall i \ \in \ \{1, \cdots, M\}, \Pr\left(\left|\hat{p}_{FAR}(\Theta_i) - p_{FAR}(\Theta_i)\right| < \epsilon\right) > 1 - \delta \\
\Longrightarrow \Pr\left(\left|\hat{p}_{FAR}(\Theta) - p_{FAR}(\Theta)\right| < \epsilon\right) > 1 - \delta.
\end{aligned} \tag{17.6}$$

This suggests, repeated use of Algorithm 17.1 would deliver an estimate $\hat{p}_{FAR}(\Theta)$ for AFAR, which satisfies (17.6). On the other hand, such way of estimating $p_{FAR}(\Theta)$ is less efficient for large M. Instead, we propose to perform the following alternative algorithm.

Consider that

$$\Pr(\mathcal{J}(\Theta) > J_{th}, \Theta = \Theta_i) = \Pr(\mathcal{J}(\Theta) > J_{th} \left| \Theta = \Theta_i \right.) \Pr(\Theta = \Theta_i).$$

The probability $p_{FAR}(\Theta)$ can be re-written as

$$p_{FAR}(\Theta) = \sum_{i=1}^{M} \Pr(\mathcal{J}(\Theta) > J_{th}, \Theta = \Theta_i).$$

Therefore, the support and PDF of Θ as the random parameter set, $\mathcal{D}_\Theta$ and D_Θ, are

$$\mathcal{D}_\Theta = \bigcup_{i=1}^{M} \left(\mathcal{D}_{\Theta_i} \cap (\Theta = \Theta_i)\right), D_\Theta(\theta) = \prod_{i=1}^{M} D_{\Theta_i}(\theta) \Pr(\Theta = \Theta_i),$$

respectively. Let

$$\mathcal{D}_{FA}(\Theta) = \{\theta, \theta \in \mathcal{D}_\Theta, \mathcal{J}(\Theta) > J_{th}\} \subset \mathcal{D}_\Theta.$$

It holds

$$p_{FAR}(\Theta) = \int_{\mathcal{D}_{FA}(\Theta)} D_\Theta(\theta) d\theta. \qquad (17.7)$$

Based on (17.7), an estimate of $p_{FAR}(\Theta)$ can be achieved using the following randomised algorithm.

Algorithm 17.2 *AFAR estimation: Given* $\epsilon \in (0, 1), \delta \in (0, 1)$,

```
• Set integer $N \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$;
• Generate $N$ samples $\theta_j, j = 1, \cdots, N,$ according to $D_\Theta(\theta)$;
• Set $n = 0$;
• For $j = 1$ to $N$
    – Compute $J(\theta_j)$ (by means of simulation);
    – If
          $\mathcal{J}(\theta_j) > J_{th},$
      set
          $n = n + 1;$
• End for
```

- *Return*

$$\hat{p}_{FAR}(\Theta) = \frac{n}{N}.$$

17.2 Fault Detection Rate and RA-aided Assessment

We now study FDR issues in the probabilistic framework using the probabilistic models for faults. We denote residual evaluation functions or test statistics in the faulty process operation and without considering model uncertainties by

$$J = \mathcal{J}\left(\Theta_f, k\right),$$

as given in the probabilistic model (16.9) for $\Theta = O$ denoting the (ideal) uncertainty-free case. Due to the possible delay in detecting a fault and considering the importance of the time instance, at which the fault is detected, J is explicitly expressed as a time function. Let k_0 be the sampling number at which the fault is present for the first time. Considering that by fault detectability study, only faulty operations are under consideration, we assume that $k_0 = 0$ without loss of generality. We call a fault being detected if the time before the first detection is within an acceptable time limit k_{stop}. In this context, a fault in pattern $\Theta_{f,i}$ is detected if

$$\exists k \in [0, k_{stop}], \text{ s.t. } \mathcal{J}\left(\Theta_{f,i}, k\right) > J_{th}.$$

Definition 17.2 *Given an FD system with the residual evaluation function or test statistic $\mathcal{J}\left(\Theta_f, k\right)$, threshold J_{th} and detection logic (17.1), we call the conditional probability*

$$p_{FDR}(\Theta_{f,i}) := Pr(\mathcal{J}\left(\Theta_f, k\right) > J_{th}, k \in [0, k_{stop}] \left|\Theta_f = \Theta_{f,i}\right.)$$

fault detection rate (FDR) w.r.t. fault pattern $\Theta_{f,i}$, and the marginal probability

$$p_{FDR}(\Theta_f) = \sum_{i=1}^{K} p_{FDR}(\Theta_{f,i}) \Pr\left(\Theta_f = \Theta_{f,i}\right)$$

average fault detection rate (AFDR).

We now address the FDR computation issues in the context of the above definition. Since the proposed RA-aided algorithms are similar to Algorithms 17.1–17.2 for the FAR computations, we only briefly describe the major steps without detailed descriptions. Denote the support and PDF of $\Theta_{f,i}$, $\mathcal{D}_{\Theta_{f,i}}$ and $D_{\Theta_{f,i}}$, by

$$\mathcal{D}_{\Theta_{f,i}} = \mathcal{D}_{\theta_{f,1}^{(i)}} \cup \cdots \cup \mathcal{D}_{\theta_{f,\beta_i}^{(i)}}, \; D_{\Theta_{f,i}} = \prod_{j=1}^{\beta_i} D_{f_j,i}\left(\theta\right),$$

and the parameter set of those detectable faults in pattern $\Theta_{f,i}$ by

$$\mathcal{D}_{FD}(\Theta_{f,i}) = \left\{\theta, \theta \in \mathcal{D}_{\Theta_{f,i}}, \mathcal{J}\left(\Theta_{f,i}, k\right) > J_{th}\right\} \subset \mathcal{D}_{\Theta_{f,i}}.$$

Correspondingly, we propose the RA-based estimations of $p_{FDR}(\Theta_{f,i})$ and $p_{FDR}(\Theta_f)$ using the randomised algorithms as follows:

$$\hat{p}_{FDR}(\Theta_{f,i}) = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}_{\mathcal{D}_{FD}}\left(\theta_j^{(i)} \left|\Theta_{f,i}\right.\right), \tag{17.8}$$

$$\mathbb{I}_{\mathcal{D}_{FD}}\left(\theta_j^{(i)} \left|\Theta_{f,i}\right.\right) = \begin{cases} 1, \text{ if } \theta_j^{(i)} \in \mathcal{D}_{FD}(\Theta_{f,i}), \\ 0, \text{ otherwise,} \end{cases} \tag{17.9}$$

$$\hat{p}_{FDR}(\Theta_f) = \sum_{i=1}^{K} \hat{p}_{FDR}(\Theta_{f,i}) \Pr\left(\Theta_f = \Theta_{f,i}\right), \tag{17.10}$$

where $\theta_j^{(i)}$, $j = 1, \cdots, N$, are i.i.d. random samples generated from $\mathcal{D}_{\Theta_{f,i}}$ according to the known PDF $D_{\Theta_{f,i}}$ and

$$N \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$$

for given ϵ, δ, which guarantees

$$\Pr\left(\left|\hat{p}_{FDR}(\Theta_{f,i}) - p_{FDR}(\Theta_{f,i})\right| < \epsilon\right) > 1 - \delta, \tag{17.11}$$

$$\Pr\left(\left|\hat{p}_{FDR}(\Theta_f) - p_{FDR}(\Theta_f)\right| < \epsilon\right) > 1 - \delta. \tag{17.12}$$

As an example, we propose the following algorithm for the realisation of $\hat{p}_{FDR}(\Theta_{f,i})$ given in (17.8).

Algorithm 17.3 *FDR estimation w.r.t. fault pattern $\Theta_{f,i}$: Given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$,*

```
• Set integer $N \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$;
• Generate $N$ samples $\theta_j^{(i)}$, $j = 1, \cdots, N$, according to $D_{\Theta_{f,i}}$;
• Set $n = 0$;
• For $j = 1$ to $N$
    – Compute $J\left(\theta_j^{(i)}\right)$ (by means of simulation)
    – If
          $\mathcal{J}\left(\theta_j^{(i)}\right) > J_{th}$,
      set
          $n = n + 1$;
• End for
```

- *Return*

$$\hat{p}_{FDR}(\Theta_{f,i}) = \frac{n}{N}.$$

17.3 Mean Time to Fault Detection and RA-aided Assessment

In practice, a process operator is often interested in the time of the *first* detection of a fault, which is in general also a random event. For our purpose, we introduce the concept of *mean time to detection of faults in a certain fault pattern* as an indicator for FD performance. Recall that not all the faults in pattern $\Theta_{f,i}$ can be detected. Since we are interested in the detection time for a detectable fault, those undetectable faults are removed from the mean time computation. We denote the probability that a *detectable fault* of pattern $\Theta_{f,i}$ is detected, for the first time, at time instant $k \in [0, k_{stop}]$ by $p_{FD}(\Theta_{f,i}, k)$.

Definition 17.3 *The expected value*

$$\rho(\Theta_{f,i}) := \mathcal{E}(k) = \sum_{k=0}^{k_{stop}} k \cdot p_{FD}(\Theta_{f,i}, k) \tag{17.13}$$

is called mean time to fault detection (MT2FD) w.r.t. fault in pattern $\Theta_{f,i}$.

It is clear that for MT2FD estimation we have to compute $p_{FD}(\Theta_{f,i}, k)$. Let $p(\Theta_{f,i}, k)$ be the probability that a fault (not necessarily detectable) of pattern $\Theta_{f,i}$ is detected for the first time at time instant $k \in [0, k_{stop}]$. Thus, the probability that a fault in pattern $\Theta_{f,i}$ is detected (in the time interval $[0, k_{stop}]$), which is also the FDR w.r.t. pattern $\Theta_{f,i}$, is given by

$$p_{FDR}(\Theta_{f,i}) = \sum_{k=0}^{k_{stop}} p(\Theta_{f,i}, k).$$

Note that

$$p_{FD}(\Theta_{f,i}, k) = \frac{p(\Theta_{f,i}, k)}{p_{FDR}(\Theta_{f,i})}. \tag{17.14}$$

That means, $p_{FD}(\Theta_{f,i}, k)$ can be computed in terms of $p(\Theta_{f,i}, k)$, $p_{FDR}(\Theta_{f,i})$. The focus of our subsequent study is on estimating $p(\Theta_{f,i}, k)$ using the RA technique, because the estimation/computation of $p_{FDR}(\Theta_{f,i})$ can be realised using Algorithm 17.3. To this end, we define the support of the random parameters of those faults, which are detected at k for the first time, by

$$\mathcal{D}_{FD,k}(\Theta_{f,i}) = \begin{cases} \left\{\theta, \theta \in \mathcal{D}_{\Theta_{f,i}}, \mathcal{J}\left(\Theta_{f,i}, 0\right) > J_{th}\right\}, \ k = 0, \\ \left\{ \begin{array}{c} \theta, \theta \in \mathcal{D}_{\Theta_{f,i}}, \mathcal{J}\left(\Theta_{f,i}, k\right) > J_{th}, \\ \mathcal{J}\left(\Theta_{f,i}, k - j\right) \le J_{th}, j = 1, \cdots, k, \end{array} \right\} k > 0. \end{cases}$$

We now propose the following randomised algorithm for estimating MT2FD, whose performance will be then analysed.

Algorithm 17.4 *MT2FD estimation:*

```
• Generate N i.i.d. random samples, $\theta_l^{(i)}, l = 1, \cdots, N$, from $\mathcal{D}_{\Theta_{f,i}}$ according to
  the known PDF $D_{\Theta_{f,i}}$;
• Set
      $n(k) = 0, k = 0, \cdots, k_{stop}$;
• For $l = 1$ to N
    – Set $k = 0$;
    – Compute $\mathcal{J}\left(\theta_l^{(i)}, k\right)$;
    – If
          $\mathcal{J}\left(\theta_l^{(i)}, k\right) > J_{th}$,
      set
          $n(k) = n(k) + 1$,
      otherwise, set
          $k = k + 1$;
    – If
          $k \le k_{stop}$,
      go to Step "Compute $\mathcal{J}\left(\theta_l^{(i)}, k\right)$"
• End for
• Compute
      $N_{FD} = \sum_{k=1}^{k_{stop}} n(k), \hat{p}_{FD}(\Theta_{f,i}, k) = \frac{n(k)}{N_{FD}}$;
• Return
      $\hat{\rho}(\Theta_{f,i}) = \sum_{k=1}^{k_{stop}} k \cdot \hat{p}_{FD}(\Theta_{f,i}, k)$.
```

It is clear that

$$\begin{aligned} n(k) &= \sum_{l=1}^{N} \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right), \\ \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right) &= \begin{cases} 1, & \text{if } \theta_l^{(i)} \in \mathcal{D}_{FD,k}(\Theta_{f,i}), \\ 0, & \text{otherwise}, \end{cases} \\ N_{FD} &= \sum_{k=1}^{k_{stop}} \sum_{l=1}^{N} \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right). \end{aligned}$$

Note that

$$\hat{p}(\Theta_{f,i}, k) = \frac{1}{N} \sum_{l=1}^{N} \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right), \hat{p}_{FDR}(\Theta_{f,i}) = \frac{N_{FD}}{N}$$

are estimates of $p(\Theta_{f,i}, k)$, $p_{FDR}(\Theta_{f,i})$, respectively. As a result, the estimate for $p_{FD}(\Theta_{f,i}, k)$ is given by

$$\hat{p}_{FD}(\Theta_{f,i}, k) = \frac{\hat{p}(\Theta_{f,i}, k)}{\hat{p}_{FDR}(\Theta_{f,i})} = \frac{1}{N_{FD}} \sum_{l=1}^{N} \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right), \tag{17.15}$$

and further MT2FD is estimated by

$$\hat{\rho}(\Theta_{f,i}) = \sum_{k=1}^{k_{stop}} k \cdot \hat{p}_{FD}(\Theta_{f,i}, k) = \sum_{k=1}^{k_{stop}} \frac{k}{N_{FD}} \sum_{l=1}^{N} \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right), \tag{17.16}$$

which is delivered by the above algorithm.

Next, we determine the necessary number N of the i.i.d. random samples. Note that $\rho(\Theta_{f,i})$ is the expectation value of the detection time instant and we are not able to apply Theorem 16.1 for the determination of N. Instead, considering that $\rho(\Theta_{f,i}) \in [0, k_{stop}]$, it is reasonable to find a lower bound for N so that for given $\epsilon \in (0, 1), \delta \in (0, 1)$,

$$\begin{gathered} \Pr\left(\frac{\left| \rho(\Theta_{f,i}) - \hat{\rho}(\Theta_{f,i}) \right|}{k_{stop}} < \epsilon \right) > 1 - \delta \Longleftrightarrow \\ \Pr\left(\left| \rho(\Theta_{f,i}) - \hat{\rho}(\Theta_{f,i}) \right| < k_{stop}\epsilon \right) > 1 - \delta. \end{gathered}$$

Since we only consider detectable faults, it is reasonable to assume that $\hat{p}_{FDR}(\Theta_{f,i}) \neq 0$. It follows from (17.14) and (17.15) that

$$\begin{aligned}
p_{FD}(\Theta_{f,i},k)-\hat{p}_{FD}(\Theta_{f,i},k) &= \frac{p_{FDR}(\Theta_{f,i})}{\hat{p}_{FDR}(\Theta_{f,i})}p_{FD}(\Theta_{f,i},k)-\hat{p}_{FD}(\Theta_{f,i},k) \\
&\quad -\frac{p_{FDR}(\Theta_{f,i})-\hat{p}_{FDR}(\Theta_{f,i})}{\hat{p}_{FDR}(\Theta_{f,i})}p_{FD}(\Theta_{f,i},k) \Longrightarrow \\
\left|p_{FD}(\Theta_{f,i},k)-\hat{p}_{FD}(\Theta_{f,i},k)\right| &\leq \left|\frac{p_{FDR}(\Theta_{f,i})}{\hat{p}_{FDR}(\Theta_{f,i})}p_{FD}(\Theta_{f,i},k)-\hat{p}_{FD}(\Theta_{f,i},k)\right| \\
&\quad +\left|\frac{p_{FDR}(\Theta_{f,i})-\hat{p}_{FDR}(\Theta_{f,i})}{\hat{p}_{FDR}(\Theta_{f,i})}p_{FD}(\Theta_{f,i},k)\right| \\
&\leq \frac{\left|p(\Theta_{f,i},k)-\hat{p}(\Theta_{f,i},k)\right|+\left|\hat{p}_{FDR}(\Theta_{f,i})-p_{FDR}(\Theta_{f,i})\right|}{\hat{p}_{FDR}(\Theta_{f,i})}.
\end{aligned} \tag{17.17}$$

Recall that according to Theorem 16.1 for

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta},$$

we have

$$\left|p(\Theta_{f,i},k)-\hat{p}(\Theta_{f,i},k)\right| < \epsilon, \left|\hat{p}_{FDR}(\Theta_{f,i})-p_{FDR}(\Theta_{f,i})\right| < \epsilon$$

with probability $1-\delta$, which results in

$$\left|p_{FD}(\Theta_{f,i},k)-\hat{p}_{FD}(\Theta_{f,i},k)\right| \leq \frac{2\epsilon}{\hat{p}_{FDR}(\Theta_{f,i})} =: \bar{\epsilon}(\Theta_{f,i}). \tag{17.18}$$

Based on (17.18), we are able to prove the following theorem, which provides us with the rule for the determination of sample number N in relationship with the estimation accuracy.

Theorem 17.1 *Given $\rho(\Theta_{f,i})$ defined in Definition 17.3, its estimate $\hat{\rho}(\Theta_{f,i})$ delivered by Algorithm 17.4 and $\bar{\epsilon}(\Theta_{f,i})$ given in (17.18) on the assumption that $\hat{p}_{FDR}(\Theta_{f,i}) \neq 0$, then for $\delta \in (0,1)$, $\epsilon \in (0,1)$ and*

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta},$$

it holds

$$\Pr\left(\frac{\left|\hat{\rho}(\Theta_{f,i})-\rho(\Theta_{f,i})\right|}{k_{stop}} < \bar{\epsilon}(\Theta_{f,i})\right) > 1-\delta. \tag{17.19}$$

Proof It follows from (17.16) that

$$\frac{\hat{\rho}(\Theta_{f,i})}{k_{stop}} = \sum_{l=1}^{N} \left(\frac{1}{N_{FD}} \sum_{k=1}^{k_{stop}} \frac{k}{k_{stop}} \mathbb{I}_{\mathcal{D}_{FD,k}} \left(\theta_l^{(i)} \left| \Theta_{f,i} \right. \right) \right) \in [0, 1] .$$

Moreover,

$$\mathcal{E} \left(\hat{\rho}(\Theta_{f,i}) \right) = \rho(\Theta_{f,i}).$$

Hence, according to Chernoff bound given in (16.15) and considering (17.18), we finally have inequality (17.19), and thus the theorem is proved.

It is evident that for a relatively small $p_{FDR}(\Theta_{f,i}), \bar{\epsilon}(\Theta_{f,i})$ can become large. In order to achieve a reliable estimation of MT2FD, ϵ should be selected sufficiently small, for instance, by means of an iterative computation.

17.4 Notes and References

This chapter has been dedicated to the realisation of the second component of our probabilistic framework: definitions and computation algorithms towards performance assessment of fault detection systems. To be specific, definitions of FAR and FDR have been re-visited and specified based on the probabilistic models introduced in the previous chapter, while the definition of MT2FD has been introduced and discussed.

Considering that model uncertainties and faults can be presented in different modes and patterns, both FAR and FDR have been defined with respect to a given mode or pattern as well as in the average sense. Correspondingly, algorithms have been provided for their computations.

The MT2FD is a random variable, whose definition has been introduced on a number of assumptions. To be close to practice applications, we have defined the time to fault detection as the time between the occurrence of a fault (of certain pattern) and the first detection of this fault. We have also assumed that only those detectable faults are under consideration. A RA-based algorithm is proposed for the MT2FD computation.

It is worth emphasising that two important parameters/conditions are included in the proposed RA-based assessment algorithms: (i) the i.i.d. samples which should be generated according to some distribution, (ii) the number of samples. The former condition is important to guarantee a correct statistic assessment of the overall operations, while the latter parameter N decides the confidential level and estimation accuracy of the applied algorithm. Unfortunately, in most of the published benchmark studies, no attention has been paid to these important points. Consequently, such studies are not representative and cannot be accepted as convincing demonstrations for the (in most cases very good) performance of the addressed methods. On the other hand, our studies suggest that a common software platform for running those proposed RAs would be very helpful for a fair assessment of any fault detection method. In fact, the RA-based definitions and the computation algorithms are independent of the fault detection methods applied. They can be realised either in the data-driven or in the model-based fashion. The process under consideration could be linear, nonlinear, static or dynamic. Motivated by this consideration, as a part of our work, great efforts have been made for the development of a MATLAB platform/tool, which is now available in the test stage. It should be pointed out that a successful application of the proposed algorithms only becomes possible if it is sufficiently supported by a software tool running on a powerful computer.

Some preliminary results of our work in this chapter have been published in [1]. The proposed RA-based assessment algorithms have also be successfully tested in research studies reported in [2, 3].

References

1. S. X. Ding, L. Li, and M. Kruger, “Application of randomized algorithms to assessment and design of observer-based fault detection systems,” *Automatica*, vol. 107, pp. 175–182, 2019.
2. M. Zhong, L. Zhang, S. X. Ding, and D. Zhou, “A probabilistic approach to robust fault detection for a class of nonlinear systems,” *IEEE Trans. on Indus. Elec.*, vol. 64, pp. 3930–3939, 2017.
3. J. Zhou, Y. Yang, Z. Zhao, and S. X. Ding, “A fault detection scheme for ship propulsion systems using randomized algorithm techniques,” *Control Engineering Practice*, vol. 81, pp. 65–72, 2018.

Chapter 18
RA-based Design of Fault Detection Systems

After building the basis of our probabilistic framework and introducing the needed computation algorithms, we are now in the position to complete the last level of the framework by developing RA-based design schemes and algorithms for fault detection systems.

18.1 Randomised Algorithms Based Threshold Settings

The major objective of introducing a threshold into a fault detection system is to reduce the FAR to an acceptable level. In the norm-based evaluation framework, a threshold is generally set as, adopting the notation introduced in our probabilistic models and used in our study presented in the previous chapter,

$$J_{th} = \sup_{\theta \in \Theta} \mathcal{J}(\theta). \tag{18.1}$$

Although this setting law leads to zero FAR, this is, unfortunately, achieved at cost of a (very) low FDR. In fact, threshold setting is a highly challenging topic both in research and applications, when uncertainties are concerned. It is state of the art in practice that threshold is set by sufficient number of repeated tests under different process operation conditions. For instance, threshold settings for sensor fault detection in Electronic Stability Program (ESP) for vehicles are optimised by huge number of driving tests under different driving maneuvers, as described in the reference given at the end of this chapter. Below, we propose two RA-based algorithms for the threshold setting, which is similar to this practice.

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_18

18.1.1 Algorithm I

For our purpose, we first apply one-sided Chernoff inequality given in Theorem 16.1, equation (16.16). According to it, for given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, if

$$N \geq \frac{1}{2\epsilon^2} \log \frac{1}{\delta}, \tag{18.2}$$

then we have

$$\Pr\left(p_{FAR}(\Theta_i) < \hat{p}_{FAR}(\Theta_i) + \epsilon\right) > 1 - \delta, \tag{18.3}$$

where, as defined and denoted in the previous chapter, $p_{FAR}(\Theta_i)$ is the real FAR w.r.t. uncertainty mode Θ_i and $\hat{p}_{FAR}(\Theta_i)$ is its estimate returned by the RA-based algorithm given below.

Algorithm 18.1 *Threshold setting Algorithm I: Given the acceptable* $FAR \in (0, 1)$ *and* $\delta \in (0, 1)$, *let* $\epsilon > 0$ *be some constant satisfying* $FAR - \epsilon > 0$ *and* $\Delta > 0$ *be the iteration tolerance.*

- *Set* $J_0 (> 0)$ *but very small, and*

  $$J_{th} = J_0;$$

- *Choose integer* N *according to (18.2);*
- *Call Algorithm 17.1 with* N *as input parameter and* n/N *as output;*
- *If*

  $$n/N \leq FAR - \epsilon,$$

  then return J_{th} *and exit,*
- *Else*

  $$J_{th} = J_{th} + \Delta,$$

  go to Step 3 (call Algorithm 17.1).

Theorem 18.1 *Algorithm 18.1 returns* J_{th} *that ensures*

$$\Pr\left\{\Pr\left(\mathcal{J}(\Theta_i) > J_{th}\right) < FAR\right\} > 1 - \delta.$$

Moreover,

$$\begin{aligned} J_{th,\min} &\leq J_{th} \leq J_{th,\min} + \Delta, \\ J_{th,\min} &= \min\left\{\bar{J}_{th} : \Pr\left(\mathcal{J}(\Theta_i) > \bar{J}_{th} \left|\Theta_f = O\right.\right) < FAR\right\}. \end{aligned} \tag{18.4}$$

Proof It is evident that n/N returned by Algorithm 17.1 is an estimate for the real false alarm rate, which satisfies, according to Theorem 16.1,

$$p_{FAR}(\Theta_i) = \Pr\left(\mathcal{J}(\Theta_i) > J_{th} \,|f = 0\right) < n/N + \epsilon$$

with probability larger than $1 - \delta$. The update of J_{th} in the last step ensures that in finite iterations, it holds

$$n/N \leq FAR - \epsilon,$$

which leads to

$$p_{FAR}(\Theta_i) < n/N + \epsilon \leq FAR.$$

That (18.4) is true follows directly from the update law of J_{th}.

It is clear that for a sufficiently small Δ, Algorithm 18.1 returns (almost) the lowest threshold,

$$J_{th} \simeq J_{th,\min},$$

by which the real false alarm rate is lower than FAR. A lower threshold means higher fault detectability. Consequently, the threshold setting by Algorithm 18.1 leads to the (almost) maximum fault detectability by simultaneously satisfying the requirement on FAR at probability larger than $1 - \delta$.

Remark 18.1 *It is worth emphasising that Algorithm 18.1 can be applied for any type of fault detection systems.*

Remark 18.2 *Algorithm 18.1 delivers a threshold that guarantees that FAR w.r.t. uncertainty mode Θ_i meets the requirement. An extension of this algorithm to deal with the threshold setting guaranteeing required AFAR is straightforward and thus is not presented.*

18.1.2 Algorithm II

Alternative to Algorithm 18.1, we propose below an algorithm for the threshold setting, which is based on Theorem 16.2.

Algorithm 18.2 *Threshold setting Algorithm II: Given ϵ as the acceptable FAR w.r.t. Θ_i and $\delta \in (0, 1)$,*

- *Set integer*

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}}; \tag{18.5}$$

- *Generate N samples $\theta_j^{(i)}, j = 1, \cdots, N,$ according to $D_{\Theta_i}(\theta)$;*
- *Set*

$$J_{th} = 0;$$

- *For $j = 1$ to N*
  - *Compute $\mathcal{J}\left(\theta_j^{(i)}\right)$ (by means of simulation),*
  - *If*

$$\mathcal{J}\left(\theta_j^{(i)}\right) > J_{th},$$

set

$$J_{th} = \mathcal{J}\left(\theta_j^{(i)}\right);$$

- *End for*
- *Return J_{th}.*

Theorem 18.2 *Algorithm 18.2 returns J_{th} that ensures*

$$\Pr\left\{\mathcal{J}(\Theta_i) > J_{th} \left| \Theta = \Theta_i, \Theta_f = O \right.\right\} \leq \epsilon.$$

Proof It is evident that J_{th} returned by Algorithm 18.2 guarantees

$$\forall \theta_j^{(i)}, j = 1, \cdots, N, \mathcal{J}\left(\theta_j^{(i)}\right) \leq J_{th}$$

or equivalently,

$$J_{th} = \max_{j=1,\cdots,N} \mathcal{J}\left(\theta_j^{(i)}\right).$$

It follows from Theorem 16.2 that for given $\epsilon \in (0, 1), \delta \in (0, 1)$, if (18.5) holds, then we have, with probability larger than $1 - \delta$,

$$\Pr\left(\mathcal{J}(\Theta_i) \leq J_{th} \left| \Theta = \Theta_i, \Theta_f = O \right.\right) \geq 1 - \epsilon \Longleftrightarrow$$
$$\Pr\left(\mathcal{J}(\Theta_i) > J_{th} \left| \Theta = \Theta_i, \Theta_f = O \right.\right) \leq \epsilon.$$

Note that $\Pr\left(\mathcal{J}(\Theta_i) > J_{th} \left| \Theta = \Theta_i, \Theta_f = O \right.\right)$ is the FAR w.r.t. Θ_i. Considering that $\frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}}$ could be much smaller than $\frac{1}{2\epsilon^2} \log \frac{1}{\delta}$, Algorithm 18.2 is more efficient from the computational point of view in comparison with Algorithm 18.1.

18.2 A RA-based Design of Observer-based Fault Detection Systems

Fault detection in uncertain dynamic systems is a challenging issue. In this section, we apply the RA-technique to dealing with this issue. We first propose a design scheme for observer-based fault detection systems and furthermore study its realisation using RA-technique.

18.2.1 Basic Idea and Problem Formulation

Consider the probabilistic process model introduced in Chap. 16

$$x(k+1) = Ax(k) + Bu(k) + E_d d(k),\, y(k) = Cx(k) + Du(k) + F_d(k),$$
$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix} + \sum_{i=1}^{l} \theta_i \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix},$$

where $x \in \mathcal{R}^n, y \in \mathcal{R}^m, u \in \mathcal{R}^{k_u}, d \in \mathcal{R}^{k_d}$ denote the process state, output, input and unknown input vectors, respectively. The system matrices A, B, C, D, E_d, F_d are of appropriate dimensions with $A_o, B_o, C_o, D_o, E_d, F_d$ denoting the known nominal system matrices. The matrix

$$\Delta := \begin{bmatrix} A_\Delta & B_\Delta \\ C_\Delta & D_\Delta \end{bmatrix} := \sum_{i=1}^{l} \theta_i \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$$

represents the polytopic uncertainty with known matrices

$$\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, i = 1, \cdots, l,$$

and random variable $\theta_i \in \left[\theta_{i,-}, \theta_{i,+}\right]$ representing possible variations of the parameter around zero.

For our purpose of residual generation, an FDF together with a post-filter of the form

$$\begin{aligned} \hat{x}(k+1) &= A_o \hat{x}(k) + B_o u(k) + L r_o(k), \\ r_o(k) &= y(k) - C_o \hat{x}(k) - D_o u(k), \\ r(z) &= R(z) r_o(z) \end{aligned}$$

is applied, where $R(z)$ is the stable post-filter and the observer gain matrix L is selected to guarantee the observer stability. In fault-free operations, the dynamics of the residual generator is governed by

$$x_r(k+1) = A_r x_r(k) + B_r u(k) + E_r d(k), \tag{18.6}$$

$$r_o(k) = C_r x_r(k) + D_\Delta u(k) + F_d d(k), \tag{18.7}$$

$$x_r(k) = \begin{bmatrix} x(k) \\ x(k) - \hat{x}(k) \end{bmatrix}, A_r = \begin{bmatrix} A_o + A_\Delta & 0 \\ A_\Delta - LC_\Delta & A_o - LC_o \end{bmatrix},$$

$$B_r = \begin{bmatrix} B_o + B_\Delta \\ B_o - LD_\Delta \end{bmatrix}, C_r = \begin{bmatrix} C_\Delta & C_o \end{bmatrix}, E_r = \begin{bmatrix} E_d \\ E_d - LF_d \end{bmatrix},$$

$$x_p(k+1) = A_p x_p(k) + B_p r_o(k), \ x_p(0) = 0, \tag{18.8}$$

$$r(k) = C_p x_p(k) + D_p r_o(k), \tag{18.9}$$

where (18.8)–(18.9) is the (minimal) state space realisation of the post-filter $R(z)$. It is assumed that the above uncertain system is stable. For the residual evaluation purpose, we use the truncated $l_{2,[k-s,k]}$ norm of the residual vector,

$$J = \|r\|^2_{2,[k-s,k]} = \sum_{i=k-s}^{k} ||r(i)||^2. \tag{18.10}$$

Now, re-write (18.6)–(18.9) as

$$r_s(k) = H_{r_o,s}\left(H_o A_r^\gamma x_r(k-s-\gamma) + H_{s+\gamma}\bar{d}_{s+\gamma}(k)\right), \tag{18.11}$$

$$r_s(k) = \begin{bmatrix} r(k-s) \\ \vdots \\ r(k) \end{bmatrix}, \bar{d}_{s+\gamma}(k) = \begin{bmatrix} u_{s+\gamma}(k) \\ d_{s+\gamma}(k) \end{bmatrix}, u_{s+\gamma}(k) = \begin{bmatrix} u(k-s-\gamma) \\ \vdots \\ u(k) \end{bmatrix},$$

$$d_{s+\gamma}(k) = \begin{bmatrix} d(k-s-\gamma) \\ \vdots \\ d(k) \end{bmatrix}, H_o = \begin{bmatrix} C_r \\ \vdots \\ C_r A_r^s \end{bmatrix}, H_{s+\gamma} = \begin{bmatrix} H_{u,s+\gamma} & H_{d,s+\gamma} \end{bmatrix},$$

$$H_{r_o,s} = \begin{bmatrix} D_p & 0 & & \\ C_p B_p & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ C_p A_p^{s-1} B_p & \cdots & C_p B_p & D_p \end{bmatrix},$$

$$\varGamma_u = H_o \begin{bmatrix} A_r^{s-1} B_r & \cdots & A_r B_r & B_r \end{bmatrix}, H_{u,s+\gamma} = \begin{bmatrix} \varGamma_u & \begin{matrix} D_\Delta & 0 & & \\ C_r B_r & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 0 \\ C_r A_r^{s-1} B_r & \cdots & \cdots & D_\Delta \end{matrix} \end{bmatrix},$$

$$\varGamma_d = H_o \begin{bmatrix} A_r^{s-1} E_r & \cdots & A_r E_r & E_r \end{bmatrix}, H_{d,s+\gamma} = \begin{bmatrix} \varGamma_d & \begin{matrix} F_d & 0 & & \\ C_r E_r & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 0 \\ C_r A_r^{s-1} E_r & \cdots & \cdots & F_d \end{matrix} \end{bmatrix}.$$

In the above equation,

$$H_o A_r^\gamma x_r(k-s-\gamma) + H_{s+\gamma}\bar{d}_{s+\gamma}(k)$$

represents the dynamics of the residual vector r_o in the time interval $[k-s, k]$, while matrix $H_{r_o,s}$ reflects the dynamics of the post-filter driven by r_o. Since the residual generator is assumed to be stable, for a large γ,

$$A_r^\gamma \approx 0,$$

which implies

$$r_s(k) = H_{r_o,s} H_{s+\gamma}\bar{d}_{s+\gamma}(k).$$

Recall that all data $r_o(i), i \in [k-s, k]$, are available for the evaluation purpose, the restriction on the structure of $H_{r_o,s}$ due to the causality of post-filter $R(z)$ can be removed. This allows us to substitute $H_{r_o,s}$ by an arbitrary matrix $\bar{W}$ and write $r_s(k)$ as

$$r_s(k) = \bar{W} H_{s+\gamma}\bar{d}_{s+\gamma}(k). \tag{18.12}$$

As a result, with evaluation function

$$J = \|r_s(k)\|^2 = \|r\|^2_{2,[k-s,k]} = \bar{d}^T_{s+\gamma}(k) H^T_{s+\gamma} W H_{s+\gamma}\bar{d}_{s+\gamma}(k)$$

the observer-based residual generator design can be now formulated as finding

$$W = \bar{W}^T \bar{W}.$$

Remark 18.3 *In fact, fault detection schemes by means of the time-frequency domain analysis of residual signals, as reported in the references given at the end of this chapter, are a special realisation of (18.12). The additional degree of design freedom thanks to $\bar{W}$ can be utilised for improving the fault detection performance.*

18.2.2 Optimal Solution and the RA Realisation Algorithm

Suppose that $H_{s+\gamma}$ is of full row rank. For a given $H_{s+\gamma}$, let

$$H_{s+\gamma} = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} V^T$$

be an SVD of $H_{s+\gamma}$, and set

$$\bar{W} = \Sigma^{-1} U^T \Rightarrow W = \left(H_{s+\gamma} H^T_{s+\gamma}\right)^{-1}. \tag{18.13}$$

It turns out for W satisfying (18.13), when $H_{s+\gamma}$ is known,

$$\|r_s(k)\|^2 \leq \left\|u_{s+\gamma}(k)\right\|^2 + \left\|d_{s+\gamma}(k)\right\|^2,$$

which allows us to set the threshold as

$$J_{th} = \left\|u_{s+\gamma}(k)\right\|^2 + \delta^2_{d,[k-s-\gamma,k]} \tag{18.14}$$

with $\delta^2_{d,[k-s-\gamma,k]}$ being the known $l_{2,[k-s-\gamma,k]}$ boundedness of $d(k)$.

Remark 18.4 *Given model (18.12), the optimal fault detection issue is in fact the standard optimal fault detection problem formulated and solved in Sect. 3.4. The above solution given in (18.13) and the threshold setting law (18.14) are identical with the solution given in Sect. 3.4 and thus optimal, when $H_{s+\gamma}$ is constant and known.*

Considering that $u_{s+\gamma}(k)$, $d_{s+\gamma}(k)$ may have (significantly) different influences on $r_s(k)$, we further propose an alternative threshold setting with different weighting for $u_{s+\gamma}(k)$, $d_{s+\gamma}(k)$. To this end, consider

$$\begin{aligned}\|r_s(k)\| &= \left\|\bar{W}H_{u,s+\gamma}u_{s+\gamma}(k) + \bar{W}H_{d,s+\gamma}d_{s+\gamma}(k)\right\| \\ &\leq \gamma_u \left\|u_{s+\gamma}(k)\right\| + \gamma_d \delta_{d,[k-s-\gamma,k]},\end{aligned} \tag{18.15}$$

where γ_u, γ_d are respectively the maximum singular value of $\bar{W}H_{u,s+\gamma}$ and $\bar{W}H_{d,s+\gamma}$. As a result, the threshold can be set as

$$J_{th} = \left(\gamma_u \left\|u_{s+\gamma}(k)\right\| + \gamma_d \delta_{d,[k-s-\gamma,k]}\right)^2. \tag{18.16}$$

It is of interest to notice that threshold setting (18.16) is less conservative than (18.14) if

$$\left(\gamma_u \left\|u_{s+\gamma}(k)\right\| + \gamma_d \delta_{d,[k-s-\gamma,k]}\right)^2 < \left\|u_{s+\gamma}(k)\right\|^2 + \delta^2_{d,[k-s-\gamma,k]}.$$

It is straightforward to prove that this is true, when

$$\left(1 - 2\gamma_u^2\right) \frac{\left\|u_{s+\gamma}(k)\right\|^2}{\delta^2_{d,[k-s-\gamma,k]}} > 2\gamma_d^2 - 1, \tag{18.17}$$

because inequality (18.17) leads to

$$\begin{aligned}\left\|u_{s+\gamma}(k)\right\|^2 + \delta^2_{d,[k-s-\gamma,k]} &> 2\left(\gamma_u^2 \left\|u_{s+\gamma}(k)\right\|^2 + \gamma_d^2 \delta^2_{d,[k-s-\gamma,k]}\right) \\ &\geq \gamma_u^2 \left\|u_{s+\gamma}(k)\right\|^2 + \gamma_d^2 \delta^2_{d,[k-s-\gamma,k]} + 2\gamma_u\gamma_d \left\|u_{s+\gamma}(k)\right\| \delta_{d,[k-s-\gamma,k]} \\ &= \left(\gamma_u \left\|u_{s+\gamma}(k)\right\| + \gamma_d \delta_{d,[k-s-\gamma,k]}\right)^2.\end{aligned}$$

Now, we are in the position to propose a RA to deal with the fault detection system design for uncertain $H_{s+\gamma}$. For the sake of simplicity, we only consider model uncertainties in mode Θ_i and denote the residual vector, the residual evaluation function and the corresponding FAR by

$$\begin{aligned} r_s(k) &= \bar{W} H_{s+\gamma}(\theta)\bar{d}_{s+\gamma}(k), \theta \in \mathcal{D}_{\Theta_i}, \\ \mathcal{J}(\Theta_i, W) &= \bar{d}^T_{s+\gamma}(k) H^T_{s+\gamma}(\theta) W H_{s+\gamma}(\theta)\bar{d}_{s+\gamma}(k), \\ p_{FAR}(\Theta_i, W) &= \Pr\left(\mathcal{J}(\Theta_i, W) > J_{th} \left| \Theta = \Theta_i, \Theta_f = O\right.\right). \end{aligned} \tag{18.18}$$

The problem to be solved for the observer-based FD system design is formulated as follows: Given $\epsilon \in (0,1)$, $\delta \in (0,1)$, J_{th} satisfying (18.14) or (18.16), solve

$$\begin{aligned} &\max_{W>0} trace(W) \\ &\text{s.t. } \Pr\left(p_{FAR}(\Theta_i, W) < \epsilon\right) \geq 1-\delta. \end{aligned} \tag{18.19}$$

Remark 18.5 *Recall that J_{th} (18.14) or (18.16) only depends on $u_{s+\gamma}(k)$, $\delta^2_{d,[k-s-\gamma,k]}$. As a result, a smaller W leads to a lower FAR and, at the same time, a lower FDR. Design problem (18.19) is a trade-off between FDR and FAR. The latter is in general limited to the acceptable level ϵ.*

We propose to apply the following algorithm to solving the above optimisation problem.

Algorithm 18.3 *Optimisation of weighting matrix W: Given $\epsilon \in (0,1)$, $\delta \in (0,1)$,*

- *Choose integer*
$$N \geq \frac{1}{2\epsilon^2}\log\frac{1}{\delta},$$
and generate N i.i.d. random samples $\theta^{(j)}$, $j = 1, \cdots, N$, according to D_{Θ_i};
- *Form $H_{s+\gamma}\left(\theta^{(1)}\right)$ and set*
$$W_1 = \left(H_{s+\gamma}\left(\theta^{(1)}\right) H^T_{s+\gamma}\left(\theta^{(1)}\right)\right)^{-1};$$
- *For $j = 2$ to N Form $H_{s+\gamma}\left(\theta^{(j)}\right)$ and solve*
$$\begin{aligned} &\min_{\Delta_j \geq 0} trace(\Delta_j) \\ &s.t. \left(H_{s+\gamma}\left(\theta^{(j)}\right) H^T_{s+\gamma}\left(\theta^{(j)}\right)\right)^{-1} - W_{j-1} + \Delta_j \geq 0, \\ &W_{j-1} - \Delta_j > 0 \end{aligned} \tag{18.20}$$
for Δ_j and set
$$W_j = W_{j-1} - \Delta_j;$$

- *End for*
- *Return*

$$W = W_N.$$

With W we set threshold either according to (18.14) or (18.16).

Theorem 18.3 *Algorithm 18.3 delivers a solution that solves*

$$\max_W trace\,(W) \tag{18.21}$$

$$s.t.\ 0 < W \leq W_j \leq \left(H_{s+\gamma}\left(\theta^{(j)}\right) H_{s+\gamma}^T\left(\theta^{(j)}\right)\right)^{-1}, \tag{18.22}$$

$$\Pr\left(p_{FAR}\left(\Theta_i, W\right) < \epsilon\right) \geq 1-\delta, \tag{18.23}$$

where $\theta^{(j)}$ is the i. i. d. random sample generated according to D_{Θ_i}, $j = 1, \cdots, N$.

Proof The proof is straightforward. Note that running the optimisation problem (18.20) in Algorithm 18.3 leads to

$$W = W_N = W_1 - \sum_{j=2}^{N} \Delta_j \Longrightarrow$$

$$trace\,(W) = trace\left(\left(H_{s+\gamma}\left(\theta^{(1)}\right) H_{s+\gamma}^T\left(\theta^{(1)}\right)\right)^{-1}\right) - \sum_{j=2}^{N} trace\left(\Delta_j\right)$$

and minimisation of $trace\left(\Delta_j\right)$ in each iteration yields furthermore

$$W_N = \arg\max_W trace\,(W)$$

subject to the constraint (18.22). Since

$$\begin{gathered}\forall j,\ H_{s+\gamma}^T\left(\theta^{(j)}\right) W_j H_{s+\gamma}\left(\theta^{(j)}\right) \leq I \Longrightarrow \\ \bar{d}_{s+\gamma}^T(k) H_{s+\gamma}^T\left(\theta^{(j)}\right) W_j H_{s+\gamma}\left(\theta^{(j)}\right) \bar{d}_{s+\gamma}(k) \\ \leq \left\|u_{s+\gamma}(k)\right\|^2 + \delta_{d,[k-s-\gamma,k]}^2 = J_{th},\end{gathered}$$

as well as

$$\begin{gathered}\bar{d}_{s+\gamma}^T(k) H_{s+\gamma}^T\left(\theta^{(j)}\right) W_j H_{s+\gamma}\left(\theta^{(j)}\right) \bar{d}_{s+\gamma}(k) \\ \leq \left(\gamma_u \left\|u_{s+\gamma}(k)\right\| + \gamma_d \delta_{d,[k-s-\gamma,k]}\right)^2 = J_{th},\end{gathered}$$

which implies for both thresholds (18.13) and (18.14),

$$\hat{p}_{FAR}(\Theta_i) = 0.$$

As a result, it follows from the discussion in the last sub-section and the one-sided Chernoff inequality given in Theorem 16.1 that for N given in Algorithm 18.3, (18.23) is satisfied.

Remark 18.6 *Alternatively, for the threshold setting Algorithm 18.2 can be applied. In that case, N can also be set equal to*

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}}.$$

18.3 A Multiple Monitoring Indices Based Fault Detection Scheme

18.3.1 Motivation and Review of Fault Detection

Adopting the notation introduced in the last two chapters, fault detection system design and implementation, both in the data-driven or model-based fashion, can be roughly described as

- building a monitoring index (test statistic or residual evaluation function) $\mathcal{J}\left(\Theta, \Theta_f\right)$,
- setting threshold J_{th} and
- running (online) detection logic

$$\mathcal{J}\left(\Theta, \Theta_f\right) \leq J_{th} \Longrightarrow \text{fault-free, otherwise faulty.} \tag{18.24}$$

In order to increase the robustness of the fault detection system against uncertainty, it is desirable that $\forall \theta \in \Theta$ and $\forall \theta_f \in \Theta_f$,

$$\begin{cases} \mathcal{J}(\theta) \leq J_{th} - \varepsilon, & \text{in the fault-free case,} \\ \mathcal{J}\left(\theta_f\right) \geq J_{th} + \varepsilon, & \text{in the faulty case,} \end{cases} \tag{18.25}$$

for some $\varepsilon > 0$. If condition (18.25) holds, the fault-free and faulty operations can be evidently separated, which allows a reliable and robust fault detection. It is worth mentioning that in this case the distance between the fault-free and faulty data sets equals to 2ε.

We observe that the use of multiple features expressed in terms of multiple monitoring indices is the common practice in machine learning aided fault diagnosis. And it could considerably improve fault detectability in comparison with a single monitoring index. This motivates us to adopt multiple monitoring indices for our fault detection purpose.

Denote the multiple monitoring indices by a vector

$$\mathcal{J}\left(\Theta,\Theta_f\right)=\begin{bmatrix}\mathcal{J}_1\left(\Theta,\Theta_f\right)\\ \vdots\\ \mathcal{J}_M\left(\Theta,\Theta_f\right)\end{bmatrix}\in\mathcal{R}^M.$$

Here, monitoring indices could be, for example, χ^2-test statistic or l_2-norm or RMS of a residual vector. They could also be frequency-time domain features of a signal. It is obvious that in case of multiple monitoring indices the fault detection logic (18.24) with a number as the threshold cannot be adopted. On the other hand, notice that the condition (18.25) can be written as

$$\begin{aligned}&\begin{cases}\mathcal{J}\left(\Theta,\Theta_f\right)-J_{th}\leq-\varepsilon: \text{fault-free case,}\\ \mathcal{J}\left(\Theta,\Theta_f\right)-J_{th}\geq\varepsilon: \text{ faulty case,}\end{cases}\\ \Longleftrightarrow&\begin{cases}\frac{-1}{\varepsilon}\mathcal{J}\left(\Theta,\Theta_f\right)+\frac{1}{\varepsilon}J_{th}\geq 1: \text{fault-free case,}\\ \frac{-1}{\varepsilon}\mathcal{J}\left(\Theta,\Theta_f\right)+\frac{1}{\varepsilon}J_{th}\leq -1: \text{faulty case,}\end{cases}\end{aligned}\tag{18.26}$$

and analogue to it, the fault detection logic (18.24) can be defined as

$$\frac{-1}{\varepsilon}\mathcal{J}\left(\Theta,\Theta_f\right)+\frac{1}{\varepsilon}J_{th}\geq 0\Longrightarrow\text{fault-free, otherwise faulty.}$$

This inspires us to introduce the following detection logic with the monitoring index vector $\mathcal{J}$

$$w\mathcal{J}+b>0\Longrightarrow\text{fault-free, otherwise faulty,}\tag{18.27}$$

and write the condition (18.26) as

$$\begin{cases}w\mathcal{J}+b\geq 1: \text{fault-free case,}\\ w\mathcal{J}+b\leq -1: \text{faulty case,}\end{cases}\tag{18.28}$$

where

$$w=\begin{bmatrix}w_1&\cdots&w_M\end{bmatrix}\in\mathcal{R}^M$$

is the weighting vector and b is some constant. It should be remarked that the forms of the decision logic (18.27) and the condition (18.28) are well-known in support vector machine (SVM) technique and widely applied for fault classification.

Note that

$$w\mathcal{J}+b=1, w\mathcal{J}+b=-1$$

are two parallel hyperplanes and the distance between them is $2/\left\|w\right\|$. Thus, in the SVM-framework, the cost function is often defined as $\left\|w\right\|$, and its minimisation yields the maximal distance between the two data sets.

According to the detection logic (18.27), the threshold setting is now formulated as finding the hyperplane

$$w\mathcal{J}+b=0,\tag{18.29}$$

which is parameterised by the weighting vector w and constant b.

18.3.2 SVM- and RA-based Design of Threshold Hyperplane

Due to the similar form, it seems likely to apply the standard SVM method to find the threshold hyperplane (18.29). On the other hand, SVM is a data-driven method that delivers the parameters w and b by means of data-based training (optimisation). Considering our intention to deal with model uncertainties and potential faults efficiently, we propose

- to apply RA-technique to generating fault-free and faulty data, and then
- to use the existing SVM algorithms to determine w and b.

Let

$$\theta^{(i)} \in \Theta, i = 1, \cdots, N_1, \theta_f^{(i)} \in \Theta_f, i = 1, \cdots, N_2$$

be the fault-free and faulty sample sets, respectively. The samples are i. i. d. and generated using a RA. Correspondingly, the monitoring index vector is denoted by

$$\mathcal{J}\left(\theta^{(i)}\right), i = 1, \cdots, N_1, \mathcal{J}\left(\theta_f^{(i)}\right), i = 1, \cdots, N_2.$$

Define

$$I_+ = \{1, \cdots, N_1\}, I_- = \{N_1 + 1, \cdots, N_1 + N_2\},$$

and correspondingly denote

$$\mathcal{J}^{(i)} = \mathcal{J}\left(\theta^{(i)}\right), i \in I_+, \mathcal{J}^{(i)} = \mathcal{J}\left(\theta_f^{(i-N_1)}\right), i \in I_-.$$

By these definitions and notations, and according to the detection logic (18.27),

$$w\mathcal{J}^{(i)} + b > 0, i \in I_- \tag{18.30}$$

indicates a miss detection. Conversely,

$$w\mathcal{J}^{(i)} + b \leq 0, i \in I_+ \tag{18.31}$$

releases a false alarm. Note that conditions (18.30) and (18.31) can also be equivalently written as

$$\exists \xi_i > 0 \text{ s.t. } w\mathcal{J}^{(i)} + b > \xi_i, i \in I_-, \tag{18.32}$$

$$\exists \xi_i > 1 \text{ s.t. } w\mathcal{J}^{(i)} + b \leq 1 - \xi_i, i \in I_+, \tag{18.33}$$

respectively. For a sufficiently large number N_1, the number

$$\hat{p}_{FAR} := \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbb{I}\left(\theta^{(i)}\right),$$
$$\mathbb{I}\left(\theta^{(i)}\right) = \begin{cases} 1, & \text{if condition (18.33) is satisfied,} \\ 0, & \text{otherwise,} \end{cases}$$

is a reliable estimate of the false alarm rate defined in Definition 17.1 (either AFAR or FAR w.r.t. a uncertainty mode). It is clear that

$$\hat{p}_{FAR} = \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbb{I}\left(\theta^{(i)}\right) \le \frac{1}{N_1}\sum_{i=1}^{N_1}\xi_i, \tag{18.34}$$
$$\xi_i \ge 0, w\mathcal{J}^{(i)} + b \le 1 - \xi_i, i \in I_+. \tag{18.35}$$

In the sequel, the number $\frac{1}{N_1}\sum\limits_{i=1}^{N_1}\xi_i$ with ξ_i satisfying (18.35) will be adopted as an indicator (upper-bound) for FAR. Similarly, the number

$$\hat{p}_{MDR} = \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbb{I}\left(\theta^{(j)}\right), \tag{18.36}$$
$$\mathbb{I}\left(\theta^{(j)}\right) = \begin{cases} 1, & \text{if condition (18.32) is satisfied,} \\ 0, & \text{otherwise,} \end{cases}$$

is an estimate of the miss detection rate (MDR), which is equal to $1 - p_{FDR}$. Note that $\hat{p}_{FAR}$, $\hat{p}_{MDR}$ given in (18.34) and (18.36) can also be equivalently expressed in terms of the so-called l_0-norm (even though it is indeed not a norm, but this term is widely used) as

$$\hat{p}_{FAR} = \frac{1}{N_1}\left\|\xi_i\right\|_0, \left\|\xi_i\right\|_0 = \sum_{i=1}^{N_1}\mathbb{I}\left(\xi_i\right), \mathbb{I}\left(\xi_i\right) = \begin{cases} 1, \xi_i \neq 0, \\ 0, \xi_i = 0, \end{cases}$$
$$w\mathcal{J}^{(i)} + b \le \xi_i, \xi_i \ge 0, i \in I_+,$$
$$\hat{p}_{MDR} = \frac{1}{N_2}\left\|\xi_j\right\|_0, \left\|\xi_j\right\|_0 = \sum_{j=1}^{N_2}\mathbb{I}\left(\xi_j\right), \mathbb{I}\left(\xi_j\right) = \begin{cases} 1, \xi_j \neq 0, \\ 0, \xi_j = 0, \end{cases}$$
$$w\mathcal{J}^{(j)} + b > \xi_j, \xi_j \ge 0, j \in I_-,$$

respectively. Now, we are in the position to formulate our optimisation problem for the determination of w, b as follows: for given acceptable $FAR = \alpha$ and

$$\mathcal{J}^{(i)} = \mathcal{J}\left(\theta^{(i)}\right), i \in I_+, \mathcal{J}^{(i)} = \mathcal{J}\left(\theta_f^{(i-N_1)}\right), i \in I_-,$$

solve the following optimisation problem

$$\min_{w,b,\xi_i,i\in I_-}\left(\frac{1}{N_2}\left\|\xi_i\right\|_0+\lambda\left\|w\right\|^2\right) \tag{18.37}$$

$$\text{s.t. } w\mathcal{J}^{(i)}+b\le\xi_i,\ \xi_i\ge 0,\ i\in I_-, \tag{18.38}$$

$$w\mathcal{J}^{(i)}+b>-\xi_i,\ \xi_i\ge 0,\ i\in I_+, \tag{18.39}$$

$$\frac{1}{N_1}\left\|\xi_i\right\|_0\le\alpha,\ i\in I_+, \tag{18.40}$$

where $\lambda>0$ is a weighting factor. The meaning of this optimisation problem is obvious. Optimisation (18.37) leads to the minimisation of the distance between the two data sets as well as MDR, which is defined by the constraint condition (18.38), while the constraint conditions (18.39)–(18.40) guarantee that FAR is bounded by given α. Knowing that the optimisation problem with l_0-norm is NP-hard, which can be approximated by an l_1-norm optimisation, we propose the following alternative optimisation problem

$$\min_{w,b,\xi_i,i\in I_-}\left(\frac{1}{N_2}\sum_{i\in I_-}\xi_i+\lambda\left\|w\right\|^2\right) \tag{18.41}$$

$$\text{s.t. } w\mathcal{J}^{(i)}+b\le\xi_i,\ \xi_i\ge 0,\ i\in I_-, \tag{18.42}$$

$$w\mathcal{J}^{(i)}+b>1-\xi_i,\ \xi_i\ge 0,\ i\in I_+, \tag{18.43}$$

$$\frac{1}{N_1}\sum_{i\in I_+}\xi_i\le\alpha,\ i\in I_+, \tag{18.44}$$

in which

$$\sum_{i\in I_-}\xi_i=\sum_{i\in I_-}\left|\xi_i\right|=\left\|\xi_i\right\|_1,$$

and conditions (18.39)–(18.40) are substituted by (18.43)–(18.44). The reason why $\left\|\xi_i\right\|_0$ in (18.40) has not been replaced by $\left\|\xi_i\right\|_1$ is that $\left\|\xi_i\right\|_0$ is not bounded by $\left\|\xi_i\right\|_1$. Instead, we know from (18.34) that

$$\frac{1}{N_1}\left\|\xi_i\right\|_0\le\frac{1}{N_1}\sum_{i=1}^{N_1}\xi_i,\ w\mathcal{J}^{(i)}+b<1-\xi_i,\ \xi_i\ge 0,\ i\in I_+,$$

and thus the requirement that FAR is bounded by α is satisfied.

We would like to emphasise that the optimisation problem (18.41)–(18.44) is a standard SVM-based two-class classification problem and there exist number of algorithms for its solution, although our formulation has been motivated by the technical demands on fault detection and derived in this context.

18.3.3 Randomised Algorithm of Designing the Threshold Hyperplane

Below is the summary of the major results for the design of the threshold superplane in form of an algorithm.

Algorithm 18.4 *Design of the threshold superplane: Given* $\epsilon \in (0,1)$, $\delta \in (0,1)$, α *and*

$$\mathcal{J} = \begin{bmatrix} \mathcal{J}_1\left(\Theta, \Theta_f\right) \\ \vdots \\ \mathcal{J}_M\left(\Theta, \Theta_f\right) \end{bmatrix} \in \mathcal{R}^M,$$

- *Choose integers*

$$N_1 \geq \frac{1}{2\epsilon^2}\log\frac{1}{\delta},\ N_2 \geq \frac{1}{2\epsilon^2}\log\frac{1}{\delta}$$

and generate N_1, N_2 *i.i.d. random samples* $\theta^{(j)}, j = 1, \cdots, N_1, \theta_f^{(i)}, i = 1, \cdots, N_2$, *according to* D_Θ, D_{Θ_f}, *respectively;*
- *Compute (by means of simulations)*

$$\mathcal{J}^{(i)} = \mathcal{J}\left(\theta^{(i)}\right), i \in I_+, \mathcal{J}^{(i)} = \mathcal{J}\left(\theta_f^{(i-N_1)}\right), i \in I_-,$$
$$I_+ = \{1, \cdots, N_1\}, I_- = \{N_1+1, \cdots, N_1+N_2\};$$

- *Solve the optimisation problem (18.41)–(18.44) for* w, b;
- *Output* w, b.

We would like to remark that $N_1 \geq \frac{1}{2\epsilon^2}\log\frac{1}{\delta}$ guarantees

$$\Pr\left(p_{FAR} < \hat{p}_{FAR} + \epsilon\right) > 1 - \delta.$$

Since

$$\hat{p}_{FAR} \leq \frac{1}{N_1}\sum_{i=1}^{N_1}\xi_i \leq \alpha, w\mathcal{J}^{(i)} + b < 1 - \xi_i, \xi_i \geq 0, i \in I_+,$$

it holds

$$\Pr\left(p_{FAR} < \alpha + \epsilon\right) > 1 - \delta.$$

Note that there is no specified requirement on the FDR $(1 - MDR)$. Hence, setting

$$N_2 \geq \frac{1}{2\epsilon^2}\log\frac{1}{\delta}$$

is in fact optional.

18.4 Benchmark Study on a Three-tank System

This section is dedicated to a benchmark study on the real laboratory three-tank system TTS20 aiming at testing

- the randomised algorithms for the FAR, FDR and MT2FD estimations given in Chap. 17,
- Algorithm 18.3 for the observer-based fault detection system design, and finally
- Algorithm 18.4 for the design of the threshold superplane.

18.4.1 System Setup and Models

A three-tank system, as sketched in Fig. 18.1, has typical characteristics of tanks, pipelines and pumps used in chemical industry and thus often serves as a benchmark process in laboratories for process control. The model and the parameters of the three-tank system introduced here are from the laboratory setup TTS20.

Applying the incoming and outgoing mass flows under consideration of Torricelli's law, the dynamics of TTS20 is modelled by

$$\begin{aligned}
&\mathcal{A}\dot{h}_1 = Q_1 - Q_{13}, \mathcal{A}\dot{h}_2 = Q_2 + Q_{32} - Q_{20}, \mathcal{A}\dot{h}_3 = Q_{13} - Q_{32}, \\
&Q_{13} = a_1 s_{13} \operatorname{sgn}(h_1 - h_3)\sqrt{2g|h_1 - h_3|}, \\
&Q_{32} = a_3 s_{23} \operatorname{sgn}(h_3 - h_2)\sqrt{2g|h_3 - h_2|}, \; Q_{20} = a_2 s_0 \sqrt{2gh_2},
\end{aligned}$$

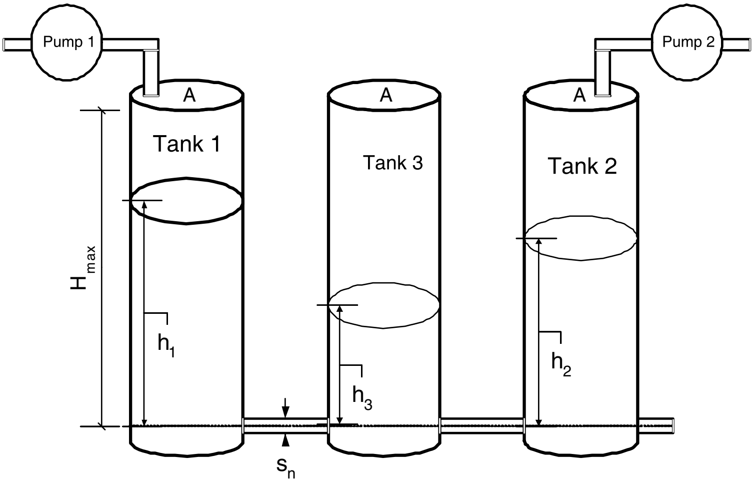


Fig. 18.1 Setup of a three-tank system

where

- Q_1, Q_2 are incoming mass flow (cm^3/s),
- Q_{ij} is the mass flow (cm^3/s) from the i-th tank to the j -th tank,
- $h_i(t)$, $i = 1, 2, 3$, are the water level (cm) in each tank and measurement variables, and
- $s_{13} = s_{23} = s_0 = s_{n.}$

The parameters are given in Table 18.1.

In most of our studies, we deal with linear systems. The linear form of the above model can be achieved by a linearisation at an operating point as follows:

$$\dot{x} = Ax + Bu, y = Cx,$$
$$x = \begin{bmatrix} h_1 - h_{1,o} \\ h_2 - h_{2,o} \\ h_3 - h_{3,o} \end{bmatrix}, u = \begin{bmatrix} Q_1 - Q_{1,o} \\ Q_2 - Q_{2,o} \end{bmatrix}, Q_o = \begin{bmatrix} Q_{1,o} \\ Q_{2,o} \end{bmatrix},$$
$$A = \frac{\partial f}{\partial h}\Big|_{h=h_o}, B = \begin{bmatrix} \frac{1}{\mathcal{A}} & 0 \\ 0 & \frac{1}{\mathcal{A}} \\ 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix},$$

where $x \in \mathcal{R}^3$ denotes the state vector with the tank levels as state variables, $u \in \mathcal{R}^2$ the two pump flows and $y \in \mathcal{R}^2$ the tank levels x_1 and x_2, $h_{i,o}$, $i = 1, 2, 3$, $Q_{1,o}$, $Q_{2,o}$ denote the operating point under consideration and

Tab. 18.1 Parameters of TTS20

Parameters	Symbol	Value	Unit
Cross section area of tanks	$\mathcal{A}$	154	cm^2
Cross section area of pipes	s_n	0.5	cm^2
Max. height of tanks	H_{max}	62	cm
Max. flow rate of pump 1	$Q_{1_{max}}$	100	cm^3/s
Max. flow rate of pump 2	$Q_{2_{max}}$	100	cm^3/s
Coeff. of flow for pipe 1	a_1	0.46	
Coeff. of flow for pipe 2	a_2	0.60	
Coeff. of flow for pipe 3	a_3	0.45	

$$f(h)=\begin{bmatrix}\frac{-a_1 s_{13}\operatorname{sgn}(h_1-h_3)\sqrt{2g|h_1-h_3|}}{\mathcal{A}}\\ \frac{a_3 s_{23}\operatorname{sgn}(h_3-h_2)\sqrt{2g|h_3-h_2|}-a_2 s_0\sqrt{2gh_2}}{\mathcal{A}}\\ \frac{a_1 s_{13}\operatorname{sgn}(h_1-h_3)\sqrt{2g|h_1-h_3|}-a_3 s_{23}\operatorname{sgn}(h_3-h_2)\sqrt{2g|h_3-h_2|}}{\mathcal{A}}\end{bmatrix}, h=\begin{bmatrix}h_1\\ h_2\\ h_3\end{bmatrix}.$$

In the steady state at the operating point, it holds

$$A\begin{bmatrix}h_{1,o}\\ h_{2,o}\\ h_{3,o}\end{bmatrix}+B\begin{bmatrix}Q_{1,o}\\ Q_{2,o}\end{bmatrix}=0\Longleftrightarrow\begin{bmatrix}A & B\end{bmatrix}\begin{bmatrix}h_o\\ Q_o\end{bmatrix}=0.$$

In TTS20 setup, it is possible to implement a nonlinear controller that leads to a full decoupling of the three tank system into

- two linear sub-systems of the first order and
- a nonlinear sub-system of the first order.

This controller can be schematically described as follows:

$$\begin{aligned}u_1&=Q_1=Q_{13}+\mathcal{A}\left(a_{11}h_1+v_1\left(w_1-h_1\right)\right),\\ u_2&=Q_2=Q_{20}-Q_{32}+\mathcal{A}\left(a_{22}h_2+v_2\left(w_2-h_2\right)\right),\end{aligned}$$

where $a_{11}, a_{22}<0$, v_1, v_2 represent two prefilters and w_1, w_2 are reference signals. The nominal (fault-free) closed-loop dynamics is described by

$$\begin{bmatrix}\dot{h}_1\\ \dot{h}_2\\ \dot{h}_3\end{bmatrix}=\begin{bmatrix}\left(a_{11}-v_1\right)h_1\\ \left(a_{22}-v_2\right)h_2\\ \frac{a_1 s_{13}\operatorname{sgn}(h_1-h_3)\sqrt{2g|h_1-h_3|}-a_3 s_{23}\operatorname{sgn}(h_3-h_2)\sqrt{2g|h_3-h_2|}}{\mathcal{A}}\end{bmatrix}+\begin{bmatrix}v_1 & 0\\ 0 & v_2\\ 0 & 0\end{bmatrix}\begin{bmatrix}w_1\\ w_2\end{bmatrix}.$$

In the steady state, we have

$$\begin{bmatrix}\left(a_{11}-v_1\right)h_1\\ \left(a_{22}-v_2\right)h_2\\ \frac{a_1 s_{13}\operatorname{sgn}(h_1-h_3)\sqrt{2g|h_1-h_3|}-a_3 s_{23}\operatorname{sgn}(h_3-h_2)\sqrt{2g|h_3-h_2|}}{\mathcal{A}}\end{bmatrix}+\begin{bmatrix}v_1 & 0\\ 0 & v_2\\ 0 & 0\end{bmatrix}\begin{bmatrix}w_1\\ w_2\end{bmatrix}=0.$$

The laboratory three-tank system TTS20 is a modified setup of the laboratory three-tank system DTS200 (see reference given at the end of this chapter). In TTS20, 6 electrical control valves are additionally installed, which enable accurate control of the flow cross section of the connection pipes. In our study, the three-tank system works around the operating point: 30 cm water level in tank 1 and 20 cm in tank 2. A linearisation at this operating point with sampling time $T_s=5$ s results in the system matrices in model (16.2) with $D_o=0$ and

$$A_o = \begin{bmatrix} 0.9305 & 0.0025 & 0.0670 \\ 0.0025 & 0.8853 & 0.0653 \\ 0.0670 & 0.0653 & 0.8660 \end{bmatrix}, B_o = \begin{bmatrix} 0.0324 & 0.0000 \\ 0.0000 & 0.0316 \\ 0.0012 & 0.0011 \end{bmatrix}.$$

18.4.2 FAR, FDR and MT2FD Assessment

In the first benchmark study on the assessment of FAR, FDR and MT2FD, uncertainties caused by the identification of the flow coefficients using experimental data are considered, which depend on the room and operation conditions. It is assumed that the variations are uniformly distributed in (0.25, 0.65), which can be further modelled as

$$\Delta_i = \theta_{\Delta_i} A_i, i = 1, \cdots, 4,$$

with

$$\theta_{\Delta_i} = \frac{\hat{\theta}_{\Delta_i}}{\sum_{j=1}^{4} \hat{\theta}_{\Delta_j}} \in [0, 1], \hat{\theta}_{\Delta_i} \sim \mathcal{U}(0, 1), i = 1, \cdots, 4,$$

representing the normalised value range of variations. Furthermore, the disturbances in the two pumps d_1, d_2 are under consideration and assumed to be

$$d_1(k) = -2 + 4 \sum_{i=0} \theta_1 \sigma(k - 5i), d_2(k) = -1 + 2 \sum_{i=0} \theta_2 \sigma(k - 5i)$$

with

$$\theta_1 \sim \mathcal{U}(0, 1), \theta_2 \sim \mathcal{U}(0, 1), \sigma(k - j) = \begin{cases} 1, k = j, \\ 0, k \neq j. \end{cases}$$

Two different types of faults are considered. The first fault is a multiplicative fault in the level sensor of tank 2 and modelled as

$$C_f = \theta_{f_1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \theta_{f_1} = \phi_1 \left(k, \theta_{\theta_{f_1},1}, \theta_{\theta_{f_1},2}\right),$$

$$\phi_1 \left(k, \theta_{\theta_{f_1},1}, \theta_{\theta_{f_1},2}\right) = \begin{cases} 0, & k < k_{fault}, \\ -\theta_{\theta_{f_1},1} \left(1 - e^{-\theta_{\theta_{f_1},2}(k - k_{fault})}\right), & k \geq k_{fault}, \end{cases}$$

$$\theta_{\theta_{f_1},1} \sim \mathcal{U}(0.02, 0.5), \theta_{\theta_{f_1},2} \sim \mathcal{U}(0.04, 0.08)$$

with k_{fault} as the time for the occurrence of the fault. The second fault is a leakage in tank 1 and modelled as

$$f_2(k) = \phi_2\left(k, \theta_{f_2}\right), \theta_{f_2} \sim \mathcal{U}(10, 30),$$

$$\phi_2\left(k, \theta_{f_2}\right) = \begin{cases} 0, k < k_{fault}, \\ 0.01\left(k - k_{fault}\right), k_{fault} \leq k \leq k_{fault} + \theta_{f_2}, \\ 0.01\theta_{f_2}, k > k_{fault} + \theta_{f_2}. \end{cases}$$

For our purpose, a linear observer-based residual generator is applied with the following observer gain matrix

$$L = \begin{bmatrix} 0.9329 \; 0.0048 \; 0.0982 \\ 0.0049 \; 0.8876 \; 0.0967 \end{bmatrix}^T .$$

Running Algorithm 18.3 for

$$\epsilon = 0.01, \delta = 10^{-7},$$

which leads to

$$N = 6905 \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}},$$

returns $W \in \mathcal{R}^{12\times 12}$ and moreover γ_u, γ_d in (18.16) are computed equal to

$$\gamma_u = 0.00934, \gamma_d = 1.$$

Next, the RA-aided assessment of FAR, FDR and MT2FD for the proposed observer-based FD system is demonstrated. With

$$\delta^2_{d,[k-s-\gamma,k]} = 40,$$

and γ_u, γ_d given above, it is checked that (18.17) is satisfied for the process operation. As a result, threshold (18.16) is adopted.

FAR Assessment

We assume that both model uncertainty and unknown input vector are simultaneously present in the three-tank system. The probabilistic parameter model for the model uncertainty and unknown input is

$$\Theta = \left\{\hat{\theta}_{\Delta_i}, i = 1, \cdots, 4, \theta_1, \theta_2\right\} := \left\{\theta_j, j = 1, \cdots, 6\right\}, D_\Theta = \bigcup_{j=1}^{6} \mathcal{U}(0, 1).$$

The assessment procedure consists of the following steps:

- By each (fault-free) simulation, u, y are first generated;

- Driven by u, y, the FD system delivers $r_s(k)$, which is then evaluated. The simulation time, which is also defined as time limit for detection, is
$$T_{sim} = \tau_{stop} = 1000\,\mathrm{s};$$
- Running Algorithm 17.1 for $\epsilon = 0.02$, $\delta = 0.001$, which gives
$$N = 9502 \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta},$$
results in
$$\hat{p}_{FAR} = 0.0132.$$

FDR and MT2FD Assessment

Let
$$\Theta_{f,1} = \left\{\theta_{f_1}\right\}, \Theta_{f,2} = \left\{\theta_{f_2}\right\}$$

denote two fault patterns for the sensor and leakage fault respectively with the probabilistic parameter model for the faults:

$$\Theta_{f,1} = \left\{\theta_{f_1}^{(1)} = \theta_{\theta_{f_1,1}}, \theta_{f_2}^{(1)} = \theta_{\theta_{f_1,2}}\right\}, \Theta_{f,2} = \left\{\theta_{f_1}^{(2)} = \theta_{f_2}\right\},$$
$$D_{f_1,1} = \mathcal{U}(0.02, 0.5), D_{f_2,1} = \mathcal{U}(0.04, 0.08), D_{f_1,2} = \mathcal{U}(10, 30).$$

For each fault pattern, the following assessment procedure is completed:

- k_{fault} is set to be 500 s;
- For every simulation with faults, u, y are first generated. The FD system delivers $r_s(k)$. The simulation time is set as $T_{sim} = 1000$ s. The time limit for detection is set equal to $k_{stop} = 50$ samplings (250 s);
- Run Algorithm 17.3 with
$$N = 9502 \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$$
for $\epsilon = 0.02$, $\delta = 0.001$. It returns:
$$\hat{p}_{FDR}(\Theta_{f,1}) = 0.9502, \hat{p}_{FDR}(\Theta_{f,2}) = 1;$$
- Applying Algorithm 17.4 for estimating MT2FD returns:
$$\hat{\rho}(\Theta_{f,1}) = 3.1630 \leq 4 \text{ samplings}, \hat{\rho}(\Theta_{f,2}) = 7.3294 \leq 8 \text{ samplings}.$$

Comparisons with Nominal Weighting Matrix

A comparison study on the fault detection performance between the proposed design approach and a fault detection approach without considering the uncertainties is performed. For this purpose, the nominal weighting matrix is set as

$$W_n = \left(H_{d,s+\gamma} H_{d,s+\gamma}^T\right)^{-1}.$$

The corresponding FAR, FDR and MT2FD are

$$\hat{p}_{FAR} = 0.3993,\ \hat{p}_{FDR}(\Theta_{f,1}) = 0.9971,\ \hat{p}_{FDR}(\Theta_{f,2}) = 1.$$

It is evident that the weighting matrix W returned by Algorithm 18.3 delivers a much better fault detection performance with a similar FDR but considerably smaller FAR (0.0132 vs. 0.3993) in comparison with a design without considering model uncertainties.

18.4.3 Fault Detection Results with Real Data

In order to demonstrate the effectiveness of the proposed fault detection scheme, the designed fault detection system is implemented for real-time fault detection. For demonstration purpose, a multiplicative fault in the level sensor of tank 2 with

$$\theta_{f_1} = 0.12$$

is realised between 520 s and 1150 s (the 104-th to 230-th samples). The detection result is shown in Fig. 18.2. For comparison, an algorithm known in the literature for the norm-based fault detection for systems with polytopic uncertainty (see the reference given at the end of this chapter) is applied, which returns a threshold

$$J_{th,2} = 0.0713\left(||u_\tau||_2 + \delta_{d,[k-\tau,k]}\right).$$

The comparison results of these two fault detection approaches are also shown in Fig. 18.2. For the comparison purpose, the result achieved using Algorithm 18.3 is labelled by “Algorithm 1”, while the result delivered by the algorithm known for the norm-based fault detection is labelled by “norm-based”. Next, a leakage in tank 1 (f_2) is realised by setting the electrical control valve at the bottom of tank 1 for

$$\theta_{f_2} = 13.$$

The leakage fault occurs at 1525 s (the 305-th sample). The detection result and its comparison with norm-based fault detection approach is given in Fig. 18.3. It is

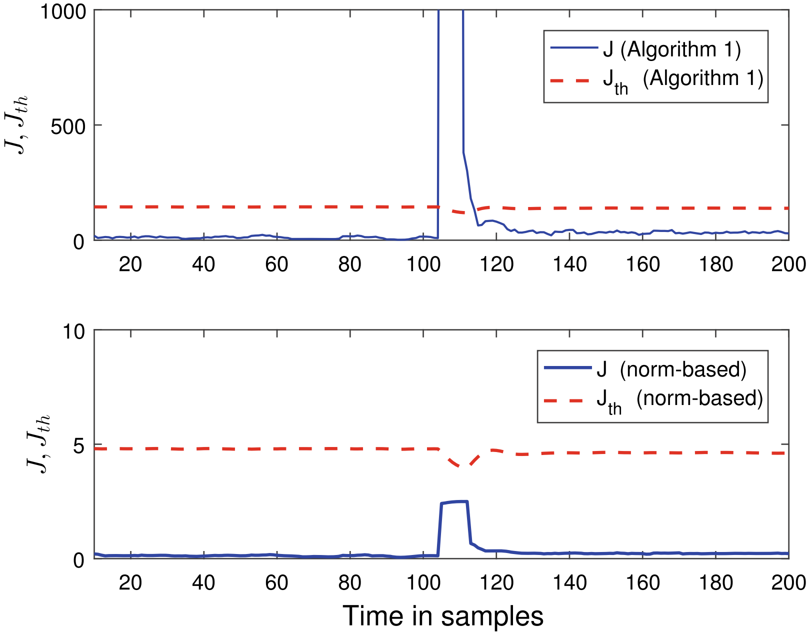


Fig. 18.2 Detection of the sensor fault

obvious that in both cases, with the weighting matrix W delivered by Algorithm 18.3 (labelled by "Algorithm 1"), the fault detectability can be significantly improved.

18.4.4 Multiple Monitoring Indices Based Fault Detection using Algorithm 18.4

In our second study, the SIMULINK model of the laboratory three-tank system TTS20 with the technical data as given in Sub-section 18.4.1 is used. In addition, measurement noises are introduced in tank 1 and tank 2, respectively, with

$$n_1 \sim \mathcal{N}(0, 0.2027), n_2 \sim \mathcal{N}(0, 0.0051).$$

In the simulation study, a nonlinear controller is used to achieve a feedback linearisation, and the total simulation time is set to be 500 s.

Using randomised algorithm, fault-free data are generated under the following conditions with parameter uncertainties and disturbances:

- the variation of the outflow coefficients $a_i, i = 1, 2$, of tank i is uniformly distributed in (0.25, 0.65), and
- for every 5 s, the disturbances in the pumps, d_1, d_2, are added with

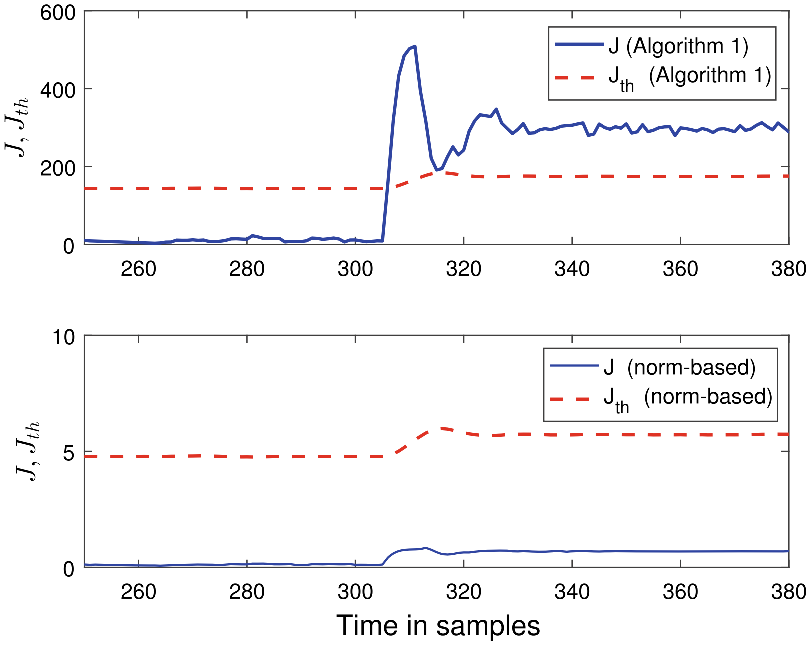


Fig. 18.3 Detection of leakage

$$\begin{aligned} d_1 &= -2 + 4\theta_1, d_2 = -1 + 2\theta_2, \\ \theta_i &\sim \mathcal{U}(0, 1), i = 1, 2. \end{aligned}$$

For the evaluation purpose, three evaluation functions are generated sequentially, which give three monitoring indices:

- the absolute value of the mean value

$$J_{average}(k) = \sqrt{\bar{r}_1^2(k) + \bar{r}_2^2(k)}$$

with $\bar{r}_i, i = 1, 2$, defined as

$$\bar{r}_i = \frac{1}{N} \sum_{j=k}^{k+N-1} r_i(j),$$

- the root mean square (RMS) value

$$J_{RMS}(k) = \sqrt{\frac{1}{N} \sum_{j=k}^{k+N-1} (r_1^2(j) + r_2^2(j))},$$

- the peak value

$$J_{peak}(k) = \max_{j\in[k,k+N-1]} \sqrt{r_1^2(j) + r_2^2(j)}.$$

By running Algorithm 18.4 with $\epsilon = 0.05$ and $\delta = 0.01$, 1060 samples are generated for the uncertainties and disturbances (in the fault-free case). After calibration, the residual signals from 201 s to 450 s (50 samples) are used for building the evaluation functions (three monitoring indices) $J_{average}(k)$, $J_{RMS}(k)$ and $J_{peak}(k)$. Due to the large amount of samples, the residual data are processed in batch with $N = 5$. Correspondingly, for each sample of the uncertainties and disturbances, we have

$$k = 1, 2, \cdots, 10.$$

That means a total sample number

$$N_1 = 10 \times 1060 = 10600.$$

In our study, the following faults are simulated:

- the sensor fault of water level in tank 2, modelled as (in percentage, 100% indicates fault-free)

$$\begin{aligned} f_1(t) &= \begin{cases} 100, t < t_{fault}, \\ 100\left(1 - \theta_{f_1,1}\left(1 - e^{-\theta_{f_1,2}(t-t_{fault})}\right)\right), t \geq t_{fault}, \end{cases} \\ \theta_{f_1,1} &\sim \mathcal{U}(0.02, 0.5), \theta_{f_1,2} \sim \mathcal{U}(0.04, 0.08), \end{aligned}$$

- the leakage fault in tank 1 with

$$\begin{aligned} f_2(t) &= \begin{cases} 0, t < t_{fault}, \\ 0.01\left(t - t_{fault}\right), t_{fault} \leq t \leq t_{fault} + \theta_{f_2}, \\ 0.01\theta_{f_2}, t > t_{fault} + \theta_{f_2}, \end{cases} \\ \theta_{f2} &\sim \mathcal{U}(10, 30), \end{aligned}$$

  where value 0 indicates fault-free.

By running Algorithm 18.4 with $\epsilon = 0.05$ and $\delta = 0.01$, 738 faulty samples are generated for each faulty scenario. It is simulated that the fault occurs at 300 s and the residual signals from 301 s to 450 s are recorded for the evaluation purpose. By the same evaluation scheme as given for the fault-free case, we have a total faulty sample number

$$N_2 = 6 \times 738 = 4428.$$

As a result, Algorithm 18.4 delivers a threshold hyperplane equal to

$$w = [-1.032 \quad -1.082 \quad -1.700], b = 1.4142.$$

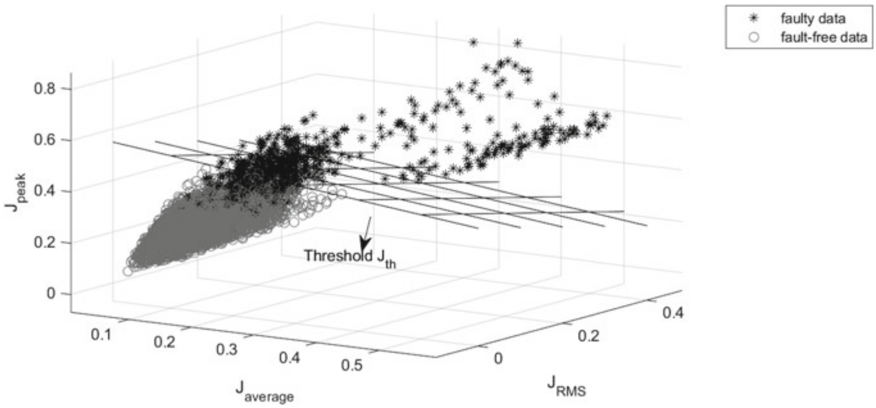


Fig. 18.4 Plot of the threshold hyperplane and the training data

In Fig. 18.4, the threshold hyperplane and the fault-free and faulty training data are plotted.

To verify the above threshold setting, two simulation examples with a (constant) sensor fault in tank 2,

$$\theta_{f_1} = 0.26,$$

and a (constant) leakage fault in tank 1 by setting the electrical control valve at the bottom of tank 1

$$\theta_{f_2} = 18.$$

Both faults occur at 5000 s and residual signals are recorded starting from 1000 s. The simulation results are shown in Figs. 18.5 and 18.6, respectively, which demonstrate a successful detection.

18.5 Notes and References

As the last component of our probabilistic framework, we have introduced, in this chapter, RA-based approaches for designing fault detection systems. The first approach has been devoted to the issue of threshold setting. Threshold is a part of any fault detection logic, and its setting is a highly challenging topic, when a trade-off between fault detectability and false alarm rate is concerned. The RA-based threshold setting algorithms proposed in this chapter can be viewed as a software- or simulation-based realisation of the common industrial practice. Analogue to the ESP example, where threshold settings for sensor fault detection are optimised by a huge number of driving tests under different driving maneuvers [1], in the RA-framework, the threshold setting is optimised using sufficient number of samples generated by

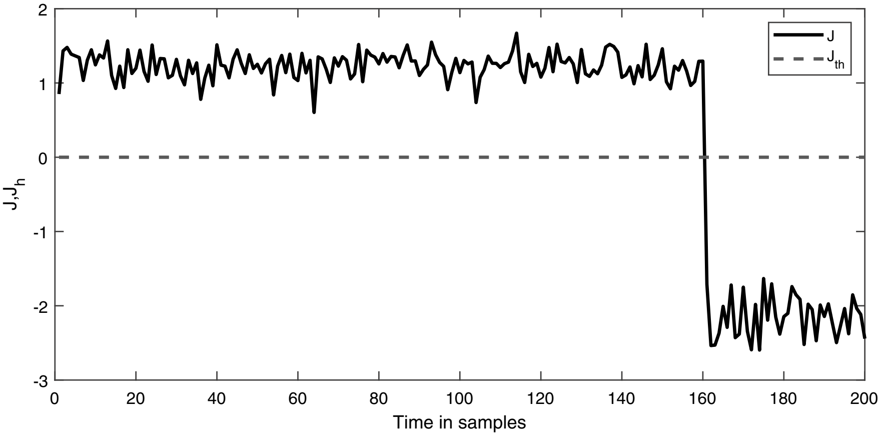


Fig. 18.5 Detection of the sensor fault

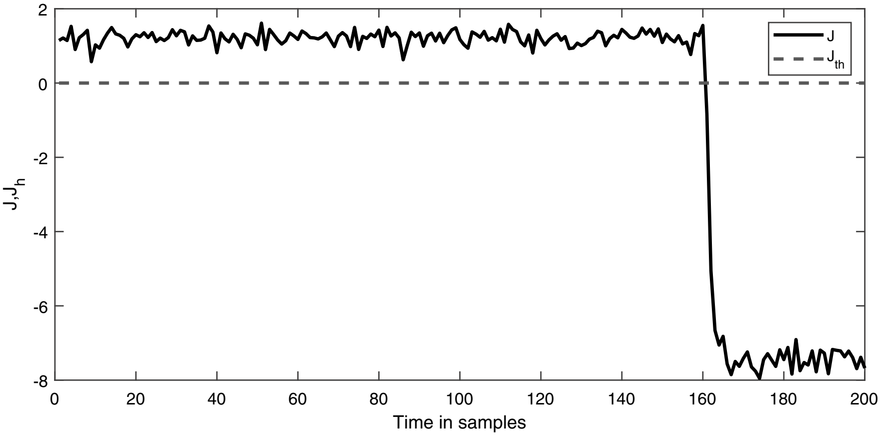


Fig. 18.6 etection of the leakage

randomised algorithms based on probabilistic parameter models. To this end, we have developed two algorithms, Algorithms 18.1 and 18.2. Both of them can be used for the threshold setting for any type of fault detection systems, for instance, in the application of Riemannian metric based fault detection schemes (with SPD matrices data sets) presented in Sub-section 15.4.2. It is worth remarking that, in comparison of both algorithms, Algorithm 18.2 is more efficient.

The second approach presented in this chapter consists of two parts. In the first part, the basic idea of an approach to the optimal design of observer-based fault detection systems (without uncertainty) has been described. It can be viewed as an extension of the fault detection schemes based on time-frequency domain analysis of residual signals, as reported in [2, 3]. To be specific, the design of a dynamic post-filter is replaced by an optimal selection of a weighting matrix. The second part of this work has been dedicated to the RA-based realisation of an optimal selection of the weighting matrix, when uncertainties modelled by the probabilistic models are under consideration. A preliminary version of Algorithm 18.3 has been reported in [4].

The last fault detection scheme addressed in this chapter is, to our knowledge, new in the model-based fault detection framework, but common in the field of machine learning aided fault diagnosis. It is the use of multiple monitoring indices. From the viewpoint of problem formulation, it seems identical with an SVM-based two-class classification problem [5]. However, it is not a classical SVM method in the sense that

- it is not a data-driven method in the classical sense. The data used in the optimisation of the threshold hyperplane are i. i. d. samples generated using RA. That means, for our purpose, process models including probabilistic models for uncertainties and faults should be available;
- the problem formulation is motivated and derived by the common specification of optimal fault detection: given an acceptable FAR, find a threshold hyperplane such that the fault detectability (FDR) is maximised. In this context, it is a natural extension of the single monitoring index based fault detection schemes to the case with multiple monitoring indices.

It can be noticed that, also for the above mentioned reasons, the variable ξ_i in our optimisation problem (18.37)–(18.40) or (18.41)–(18.44) has been introduced in the context of miss detection and false detection, which is different from the use as the so-called slack variable in the SVM-framework for the optimisation purpose (in sense of soft margin) [6]. Moreover, the use of the so-called l_0-norm in the original problem formulation (18.37)–(18.40) is the result of the RA-based MDR and FAR computation, rather than for achieving sparsity as its objective. On the other hand, knowing the optimisation problem with l_0-norm is NP-hard, but can be approximated by an l_1-norm optimisation [7], we have proposed an alternative optimisation problem (18.41)–(18.44). It should be emphasised that the constraint (18.44) plays an important role in this optimisation problem, which meets the demand for the FAR.

Thanks to the identical optimisation problem formulation, the existing optimisation algorithms commonly used in the SVM-technique can be directly applied for our problem solution [6, 8].

We would like to provide background information about this work, which is a part of the DFG (German Research Foundation) project entitled "Application of randomized algorithms to the analysis and synthesis of model-based and data-driven fault diagnosis systems". During a project workshop in year 2016, Dr. Mingzhu Tang from the Changsha University of Science & Technology, P.R. China, has presented application of the advanced SVM-algorithm CLDM (cost-sensitive largemargin distribution machine) to fault diagnosis [9]. His presentation and the follow-up discussions have inspired the idea of applying RA-technique to multiple monitoring indices based fault detection and formulating the associated optimal fault detection problems with the help of SVM-formalism. This idea and the established problem formulation have been later presented at Peking University, P.R. China. Mr. Zhou, at that time a Master student of that university, has then applied these results to fault detection in ship propulsion systems and reported the ideas in form of a paper with a focus on the SVM interpretation of the ideas [10].

In our benchmark study, the laboratory three-tank system TTS20 has been used. It is an update of three-tank system DTS200, which is well described in [11]. In our comparison study using real test data in Sub-section 18.4.3, the result achieved by means of Algorithm 18.3 proposed in this chapter has been compared with the ones delivered by Algorithm 9.2 given in [11], which has been proposed for designing norm-based fault detection for systems with polytopic uncertainty.

References

1. E. L. Ding, H. Fennel, and S. X. Ding, "Model-based diagnosis of sensor faults for ESP systems," *Control Engineering Practice*, vol. 12, pp. 847–856, 2004.
2. H. Ye, S. X. Ding, and G. Wang, "Integrated design of fault detection systems in time-frequency domain," *IEEE Trans. on Automatic Control*, vol. 47(2), pp. 384–390, 2002.
3. T. Xue, M. Zhong, S. X. Ding, and H. Ye, "Stationary wavelet transform aided design of parity space vectors for fault detection in LDTV systems," *IET Control Theory and Applications*, vol. 12, pp. 857–864, 2018.
4. S. X. Ding, L. Li, and M. Kruger, "Application of randomized algorithms to assessment and design of observer-based fault detection systems," *Automatica*, vol. 107, pp. 175–182, 2019.
5. V. N. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, Inc, 1998.
6. C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. on Neural Networks*, vol. 13, pp. 415–425, 2002.
7. C. Ramires, V. Kreinovich, and M. Argaez, "Why l_1 is a good approximation to l_0: A geometric explanation," *Journal of Uncertain Systems*, vol. 7, 2013.
8. M. A. Davenport, R. G. Baraniuk, and C. D. Scott, "Tuning support vector machines for minimax and neyman-pearson classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1888–1898, 2010.
9. M. Tang, S. X. Ding, C. Yang, F. Cheng, Y. Shardt, W. Long, and D. Liu, "Cost-sensitive large margin distribution machine for fault detection of wind turbines," *Cluster Computing*, 2018.
10. J. Zhou, Y. Yang, S. X. Ding, Y. Zi, and M. Wei, "A fault detection and health monitoring scheme for ship propulsion systems using SVM technique," *IEEE Access*, vol. 6, pp. 16 207–16 215, 2018.
11. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.

Part VI
An Integrated Framework of Control and Diagnosis, and Fault-tolerant Control Schemes

Chapter 19
Residual Centered Control and Detection

In our review on feedback control systems in Chap. 5, a so-called fault-tolerant control architecture has been introduced, whose core is an observer as well as an observer-based residual generator. In this context, we have claimed that each dynamic (output) feedback controller is driven by the residual signal and can be understood as an estimator for a state feedback law. Our further work on fault detection in feedback control loops in Chap. 9 has revealed that maximising fault detectability and maximising stability margin can be achieved in a unified manner. This result is in contradiction to the common consensus that a good (feedback) controller reduces the fault detectability.

In this chapter, we will study some of these issues from the perspective of system performance as well as performance degradation caused by faults. This work will unify control and detection studies and establish the fundament for our subsequent work on fault-tolerant control and performance degradation recovery, in which residual signals play a central role.

19.1 Residuals and Residual-based Unified System Models

Suppose that the nominal (fault- and uncertainty-free) system under consideration is an LTI system of the form

$$y(z) = G(z)u(z), \, y \in \mathcal{C}^m, \, u \in \mathcal{C}^p, \tag{19.1}$$

whose minimal state space realisation is given by

$$\mathcal{G} : x\,(k+1) = Ax\,(k) + Bu(k), x(0) = x_0, \, y(k) = Cx(k) + Du(k). \tag{19.2}$$

Here, $x \in \mathcal{R}^n$ is the state vector, and A, B, C, D are system matrices of appropriate dimensions. Let

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_19

$$\hat{x}\,(k+1) = A\hat{x}\,(k) + Bu(k) + L\left(y(k) - \hat{y}(k)\right), \tag{19.3}$$

$$r(k) = y(k) - \hat{y}(k), \hat{y}(k) = C\hat{x}(k) + Du(k) \tag{19.4}$$

be a state observer as well as an observer-based residual generator. It is remarkable that system (19.3)–(19.4) can be re-written as

$$\hat{x}\,(k+1) = A\hat{x}\,(k) + Bu(k) + Lr(k), \tag{19.5}$$

$$y(k) = r(k) + C\hat{x}(k) + Du(k), \tag{19.6}$$

whose input-output dynamics is fully identical with the one of the nominal system (19.1). In fact, for

$$\hat{x}\,(0) = x_0,$$

we have

$$\hat{x}\,(k) = x(k)$$

as well. In other words, the dynamic system described by (19.5)–(19.6) is an alternative input-output model for the dynamics of the nominal system (19.1).

Definition 19.1 *The dynamic system (19.5)–(19.6) is called observer-based input-output model of the nominal system (19.1).*

We now consider feedback control systems. Recall that any feedback controller can be parameterised by

$$u(z) = F\hat{x}\,(z) - Q(z)r(z),\; Q(z) \in \mathcal{RH}_\infty. \tag{19.7}$$

It is evident that, different from the nominal model (19.1), all variables needed for realising a feedback controller given in (19.7) are available in the observer-based input-output model (19.5)–(19.6). This feature is of essential importance for our subsequent work.

Using the model (19.5)–(19.6), it is straightforward (see also our discussion in Sect. 5.2) to find out that, for

$$u(z) = F\hat{x}\,(z) - Q(z)r(z) + v(z) \tag{19.8}$$

with v as a reference signal, it holds

$$\begin{bmatrix} u\,(z) \\ y\,(z) \end{bmatrix} = \begin{bmatrix} -\hat{Y}\,(z) - M\,(z)\,Q\,(z) \\ \hat{X}\,(z) - N\,(z)\,Q\,(z) \end{bmatrix} r\,(z) + \begin{bmatrix} M\,(z) \\ N\,(z) \end{bmatrix} v\,(z)\,. \tag{19.9}$$

Equation (19.9) is the closed-loop model of the nominal system (19.1) with the transfer matrix pairs $(M(z), N(z))$ and $\left(\hat{X}(z), \hat{Y}(z)\right)$ given by

$$M(z) = (A + BF, B, F, I)\,, N(z) = (A + BF, B, C + DF, D)\,, \quad (19.10)$$
$$\hat{X}(z) = (A + BF, L, C + DF, I)\,, \hat{Y}(z) = (A + BF, -L, F, 0)\,. \quad (19.11)$$

Remark 19.1 *It should be remarked that the control law of the feedback control loop configuration shown in Fig. 9.1,*

$$u(z) = K(z)y(z) + v(z), \quad (19.12)$$

is slightly different from the one given in (19.8). The difference lies in the feed-forward controller. This can be illustrated as follows. According to the Youla parameterisation, the feedback gain matrix K is written as

$$K(z) = -\left(X(z) - Q(z)\hat{N}(z)\right)^{-1}\left(Y(z) + Q(z)\hat{M}(z)\right),$$

where

$$\hat{M}(z) = (A - LC, -L, C, I)\,, \hat{N}(z) = (A - LC, B - LD, C, D)\,,$$
$$X(z) = (A - LC, -(B - LD), F, I)\,, Y(z) = (A - LC, -L, F, 0)\,.$$

It yields

$$\begin{aligned} &X(z)u(z) + Y(z)y(z) + Q(z)\left(\hat{M}(z)y(z) - \hat{N}(z)u(z)\right) \\ &= \left(X(z) - Q(z)\hat{N}(z)\right)v(z). \end{aligned}$$

Writing the observer as

$$\hat{x}\,(k+1) = (A - LC)\,\hat{x}\,(k) + (B - LD)\,u(k) + Ly(k)$$

makes it clear that

$$X(z)u(z) + Y(z)y(z) = u(z) - F\hat{x}\,(z)\,.$$

As a result, the observer-based realisation of the control law (19.12) in the loop configuration shown in Fig. 9.1 is

$$u(z) = F\hat{x}\,(z) - Q(z)r(z) + \left(X\,(z) - Q(z)\hat{N}\,(z)\right)v(z),$$

which demonstrates the difference to the control law (19.8) in the term with the reference signal v. Since our study is focusing on the influences of system uncertainties and faults, we view

$$\bar{v}\,(z) = \left(X\,(z) - Q(z)\hat{N}\,(z)\right)v(z)$$

as a reference signal and thus neglect this difference without loss of generality.

We now study the influences of unknown inputs, parameter uncertainties and faults on the system dynamics. For our purpose, we will not distinguish between faults and disturbances, instead, summarise them as system uncertainties. We adopt the model

$$x(k+1) = Ax(k) + Bu(k) + E_d d(k), y(k) = Cx(k) + Du(k) + F_d d(k) \quad (19.13)$$

to represent additive uncertainties (unknown input vector) with $d(k) \in \mathcal{R}^{k_d}$ and E_d, F_d being known system matrices of appropriate dimensions. The dynamics of the residual vector r with respect to d is governed by

$$e(k+1) = (A - LC)e(k) + (E_d - LF_d)d(k), r(k) = Ce(k) + F_d d(k),$$

where

$$e(k) = x(k) - \hat{x}(k).$$

In the case of coprime factor uncertainties, for instance, modelled by

$$y(z) = G(z)u(z) = \left(\hat{M}(z) + \Delta_{\hat{M}}\right)^{-1}\left(\hat{N}(z) + \Delta_{\hat{N}}\right)u(z), \quad (19.14)$$

the dynamics of the residual vector r is described by

$$r(z) = \Delta_{\hat{N}}u(z) - \Delta_{\hat{M}}y(z).$$

In the above model, the transfer matrix pair $\left(\hat{M}(z), \hat{N}(z)\right)$ is the LCF pair of the nominal system (19.1) and

$$\begin{bmatrix} \Delta_{\hat{M}} & \Delta_{\hat{N}} \end{bmatrix} \in \mathcal{H}_\infty$$

denotes (stable) uncertainty. Recalling the relations between the left and right coprime factor uncertainties, as given in Chap. 9, it is sufficient to consider the above left coprime factor uncertainty without loss of generality.

A further class of uncertainties is modelled by

$$x(k+1) = Ax(k) + Bu(k) + Ew(k), \quad (19.15)$$
$$y(k) = Cx(k) + Du(k) + Fw(k), \quad (19.16)$$
$$d(k) = C_d x(k) + D_d u(k) + F_d w(k), \quad (19.17)$$
$$w(z) = \Delta_d d(z), \Delta_d \in \mathcal{H}_\infty, \quad (19.18)$$

where $d \in \mathcal{R}^{k_d}, w \in \mathcal{R}^{k_w}$ and E, F, C_d, D_d, F_d are known system matrices of appropriate dimensions. Δ_d is stable but unknown, and represents system uncertainty. It is the LFT model. It yields

$$\begin{aligned} y(z) &= G_{yu}(z)\,u(z) + G_{yw}(z)w(z), \\ d(z) &= G_{du}(z)u(z) + G_{dw}(z)w(z),\, w(z) = \varDelta_d d(z), \\ G_{yu}(z) &= G(z) = (A, B, C, D)\,, G_{yw}(z) = (A, E, C, F)\,, \\ G_{du}(z) &= (A, B, C_d, D_d)\,, G_{dw}(z) = (A, E, C_d, F_d)\,, \end{aligned}$$

which leads to

$$y(z) = \left(G_{yu}(z) + G_{yw}(z)\varDelta_d\,(I - G_{dw}(z)\varDelta_d)^{-1}\,G_{du}(z)\right)u(z). \tag{19.19}$$

Here, it is assumed that $I - G_{dw}(z)\varDelta_d$ is invertible. Recall that the observer-based residual generator (19.3)–(19.4) can be written as

$$\begin{aligned} r(z) &= \hat{M}(z)y(z) - \hat{N}(z)u(z), \\ \hat{M}(z) &= (A - LC, -L, C, I)\,, \hat{N}(z) = (A - LC, B - LD, C, D)\,. \end{aligned}$$

We have the dynamics of the residual generator as

$$\begin{aligned} r(z) &= \hat{N}_w(z)\varDelta_d\,(I - G_{dw}(z)\varDelta_d)^{-1}\,G_{du}(z)u(z), \\ \hat{N}_w(z) &= (A - LC, E - LF, C, F)\,, \end{aligned}$$

which, on the assumption that

$$\hat{N}_w(z)\varDelta_d\,(I - G_{dw}(z)\varDelta_d)^{-1}\,G_{du}(z) \in \mathcal{H}_\infty,$$

is stable. In the sequel, this class of uncertainty will be addressed as a special case of the left coprime factor uncertainty with

$$\begin{bmatrix} \varDelta_{\hat{M}} & \varDelta_{\hat{N}} \end{bmatrix} = \begin{bmatrix} 0 & \hat{N}_w(z)\varDelta_d\,(I - G_{dw}(z)\varDelta_d)^{-1}\,G_{du}(z) \end{bmatrix} \in \mathcal{H}_\infty.$$

It is remarkable that all these types of uncertainties affect the model parameters or/and the dynamics of the system under consideration directly. In general, they are not measurable and accessible. On the contrary, the existence of the system uncertainties will not cause any change in the observer-based input-output model (19.5)–(19.6) explicitly. Information about the uncertainties is fully embedded in the residual vector, as demonstrated in the above study, which is available and accessible in the model (19.5)–(19.6). These two different model forms for the same system under consideration are schematically demonstrated in Fig. 19.1, in which $\varDelta$ is used to denote system uncertainties.

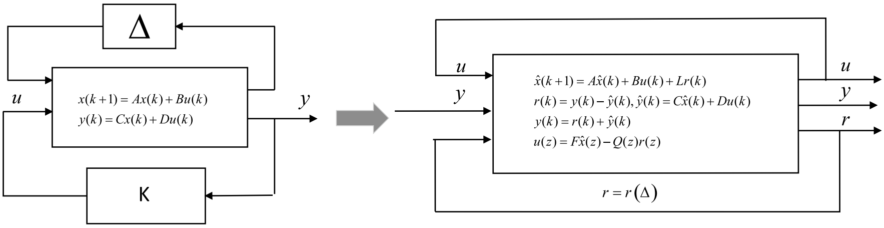

Δ
u
$x(k+1) = Ax(k) + Bu(k)$
$y(k) = Cx(k) + Du(k)$
y
K
u
y
$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr(k)$
$r(k) = y(k) - \hat{y}(k), \hat{y}(k) = C\hat{x}(k) + Du(k)$
$y(k) = r(k) + \hat{y}(k)$
$u(z) = F\hat{x}(z) - Q(z)r(z)$
u
y
r
$r = r(\Delta)$

Fig. 19.1 From the standard model to the observer-based I/O-model: a schematic description

19.2 Control Performance Degradation, Assessment and Monitoring

Loop transfer recovery (LTR) is a classic topic of control theory. Roughly speaking, LTR deals with recovering control performance degradation caused by the use of the state estimate, instead of the state variables themselves, in a state feedback controller. In this section, we are going to extend this concept to the assessment and monitoring of system performance degradation in a more general context.

19.2.1 Loop Performance Degradation

For our purpose, we first define the ideal (reference) system performance. Consider the nominal model (19.2) and re-write it as

$$x_{ideal}\left(k+1\right) = Ax_{ideal}\left(k\right) + Bu_{ideal}(k), \tag{19.20}$$

$$y_{ideal}\left(k\right) = Cx_{ideal}\left(k\right) + Du_{ideal}(k). \tag{19.21}$$

With $x_{ideal}\left(k\right)$, $y_{ideal}(k)$ we denote the ideal state and output variables, respectively, which are decoupled from any uncertainty. Moreover, we define

$$u_{ideal}(k) := Fx_{ideal}\left(k\right) + v(k). \tag{19.22}$$

Recall that

$$u(z) = F\hat{x}\left(z\right) - Q(z)r(z) + v(z)$$

is the parameterisation form of the feedback controller and $F\hat{x}\left(z\right) - Q(z)r(z)$ is an estimate for Fx_{ideal} as well. Thus, the difference $e_u(z)$,

$$e_u(z) = u_{ideal}(z) - u(z) = Fe_x(z) + Q(z)r(z),$$
$$e_x(z) = x_{ideal}(z) - \hat{x}(z),$$

quantifies the (performance) loss in the control signal caused by the use of the estimate for the state vector x. Similarly, we define the difference,

$$\begin{aligned} e_y(z) &= y_{ideal}(z) - y(z) = y_{ideal}(z) - \hat{y}(z) - r(z) \\ &= (C + DF)\, e_x(z) + DQ(z)r(z) - r(z), \end{aligned}$$

that indicates the loss in the output (performance). Using the observer-based input-output model (19.5)–(19.6) yields

$$\begin{aligned} e_x(z) &= (zI - A_F)^{-1} (BQ(z) - L)\, r(z),\ A_F = A + BF, \\ e_u(z) &= F\,(zI - A_F)^{-1} (BQ(z) - L)\, r(z) + Q(z)r(z), \\ e_y(z) &= (C + DF)\,(zI - A_F)^{-1} (BQ(z) - L)\, r(z) + DQ(z)r(z) - r(z), \end{aligned}$$

which can be further written as, noting (19.10)–(19.11),

$$e_{LPD}(z) = \begin{bmatrix} e_u(z) \\ e_y(z) \end{bmatrix} = \begin{bmatrix} \hat{Y}(z) + M(z)Q(z) \\ -\hat{X}(z) + N(z)Q(z) \end{bmatrix} r(z). \tag{19.23}$$

We call system (19.23) (control) loop performance degradation model (LPDM) with the residual vector r as input and e_{LPD} as output. Note that $e_{LPD}(z)$ can be (online) computed using either (19.23) or

$$e_{LPD}(z) = \begin{bmatrix} M\,(z) \\ N\,(z) \end{bmatrix} v\,(z) - \begin{bmatrix} u\,(z) \\ y\,(z) \end{bmatrix}.$$

Consequently, $e_{LPD}(z)$ can be understood as the difference between the SIR of the nominal system (the ideal state feedback control case) and the real system input and output signals $(u(z), y(z))$.

We would like to emphasise that the dynamic system in the LPDM,

$$\begin{bmatrix} \hat{Y}(z) + M(z)Q(z) \\ -\hat{X}(z) + N(z)Q(z) \end{bmatrix},$$

is the SIR of the feedback controller used. Thus, tuning the controller can directly reduce the (control) loop performance degradation (LPD) caused by the model uncertainties. Moreover, as discussed in Chap. 9, both the controller and observer design can change the dynamics of the residual generator with respect to the model uncertainties.

19.2.2 Assessment and Monitoring of Control Performance Degradation

In this and next sub-sections, we will propose two performance degradation assessment and monitoring schemes. The first one is dedicated to the assessment of control performance with respect to an ideal linear quadratic (LQ) controller (regulator). The second one addresses the performance assessment issue based on loop data e_{LPM} with respect to any given controller.

Ideal LQ Controller

For our purpose, we design an LQ controller as the ideal controller for the nominal system (19.20). Let the cost function be

$$J(i) = \sum_{k=i}^{\infty} \gamma^{k-i} \left[x_{ideal}^T(k) \; u_{ideal}^T(k) \right] \begin{bmatrix} Q_x & Q_{xu} \\ Q_{ux} & Q_u \end{bmatrix} \begin{bmatrix} x_{ideal}(k) \\ u_{ideal}(k) \end{bmatrix}, \tag{19.24}$$
$$\begin{bmatrix} Q_x & Q_{xu} \\ Q_{ux} & Q_u \end{bmatrix} \geq 0, \, Q_u > 0, \, Q_{ux} = Q_{xu}^T, 0 < \gamma \leq 1.$$

Applying dynamic programming technique to solving the LQ optimisation problem,

$$\min_{u_{ideal}} J(i)$$
$$\text{s.t. } x_{ideal}\,(k+1) = Ax_{ideal}\,(k) + Bu_{ideal}(k),$$

results in

$$\begin{aligned} \min_{u_{ideal}} J(i) &= x_{ideal}^T(i) P x_{ideal}\,(i)\,, \\ u_{ideal}^*(i) &= \arg\min_{u_{ideal}} J(i) = Kx_{ideal}\,(i)\,, \\ K &= -\left(Q_u + \gamma B^T PB\right)^{-1} \left(Q_{ux} + \gamma B^T PA\right), \end{aligned}$$

with

$$\begin{aligned} P =& \gamma A^T PA + Q_x + K^T \left(Q_u + \gamma B^T PB\right) K \\ &+ K^T \left(Q_{ux} + \gamma B^T PA\right) + \left(\gamma A^T PB + Q_{xu}\right) K \Longleftrightarrow \\ P =& \gamma A^T PA + Q_x - K^T \left(Q_u + \gamma B^T PB\right) K \Longleftrightarrow \\ P =& \gamma A_K^T PA_K + Q_x + K^T Q_u K + K^T Q_{ux} + Q_{xu} K, \\ A_K =& A + BK, \, P > 0. \end{aligned}$$

Reference System

We consider a reference system described by

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k), \tag{19.25}$$
$$y(k) = C(k)x(k) + D(k)u(k) + q(k), \tag{19.26}$$

where $w(k), q(k)$ are process and measurement noise vectors. It is assumed that they are uncorrelated with the state and input vectors, and

$$w(k) \sim \mathcal{N}\left(0, \Sigma_w\right), q(k) \sim \mathcal{N}\left(0, \Sigma_q\right), \tag{19.27}$$
$$\mathcal{E}\left(\begin{bmatrix} w(i) \\ q(i) \\ x(0) \end{bmatrix} \begin{bmatrix} w(j) \\ q(j) \\ x(0) \end{bmatrix}^T\right) = \begin{bmatrix} \begin{bmatrix} \Sigma_w & S_{wq} \\ S_{wq}^T & \Sigma_q \end{bmatrix} \delta_{ij} & 0 \\ 0 & \Pi_0 \end{bmatrix}$$

with known matrices $\Sigma_w, \Sigma_q, S_{wq}$. Applying a (steady) Kalman filter for the state estimation and residual generation purposes leads to

$$\hat{x}(k+1\,|k) = A\hat{x}(k\,|k-1) + Bu(k) + Lr(k), \hat{x}(0) = 0, \tag{19.28}$$
$$r(k) = y(k) - \hat{y}(k\,|k-1), \hat{y}(k\,|k-1) = C\hat{x}(k\,|k-1) + Du(k), \tag{19.29}$$
$$Y = AYA^T + \Sigma_w - L\Sigma_r L^T, L = \left(AYC^T + S_{wq}\right)\Sigma_r^{-1},$$
$$Y = \mathcal{E}\left(\left(x(k) - \hat{x}(k\,|k-1)\right)\left(x(k) - \hat{x}(k\,|k-1)\right)^T\right),$$
$$\Sigma_r = \mathcal{E}\left(r(k)r^T(k)\right) = CYC^T + \Sigma_q, Y > 0$$

with the residual vector $r(k) \in \mathcal{R}^m$ being white and L as the Kalman filter (observer) gain matrix.

Since our reference system is a stochastic process, the cost function under consideration is modified to

$$J_R(i) = \mathcal{E}\sum_{k=i}^{\infty} \gamma^{k-i} \left[x^T(k)\ u^T(k)\right] \begin{bmatrix} Q_x & Q_{xu} \\ Q_{ux} & Q_u \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \tag{19.30}$$
$$\begin{bmatrix} Q_x & Q_{xu} \\ Q_{ux} & Q_u \end{bmatrix} \geq 0, Q_u > 0, Q_{ux} = Q_{xu}^T, 0 < \gamma < 1.$$

Minimising $J_R(i)$ is the well-established linear quadratic Gaussian (LQG) control problem. Below, we solve this optimisation problem alternatively based on the observer-based input-output model (19.5)–(19.6) with

$$\hat{x}(k) = \hat{x}(k\,|k-1), \hat{y}(k) = \hat{y}(k\,|k-1).$$

Let

$$\min_{u} J_R(i) = \hat{x}^T(i\,|i-1\,)P\hat{x}(i\,|i-1\,) + c$$

and write the minimisation of $J_R(i)$ as, following the dynamic programming method,

$$\begin{aligned}
\min_{u} J_R(i) &= \min_{u(i)} \mathcal{E} \left(\begin{array}{c} \begin{bmatrix} x^T(i) \; u^T(i) \end{bmatrix} \begin{bmatrix} Q_x & Q_{xu} \\ Q_{ux} & Q_u \end{bmatrix} \begin{bmatrix} x(i) \\ u(i) \end{bmatrix} \\ +\gamma \hat{x}^T(i+1\,|i\,)P\hat{x}(i+1\,|i\,) + \gamma c \end{array} \right) \\
&= \min_{u(i)} \mathcal{E} \left(\begin{array}{c} \begin{bmatrix} x^T(i) \; u^T(i) \end{bmatrix} \begin{bmatrix} Q_x & Q_{xu} \\ Q_{ux} & Q_u \end{bmatrix} \begin{bmatrix} x(i) \\ u(i) \end{bmatrix} \\ +\gamma \left(A\hat{x}(i\,|i-1\,) + Bu(i) + Lr(i) \right)^T P \\ \cdot \left(A\hat{x}(i\,|i-1\,) + Bu(i) + Lr(i) \right) + \gamma c \end{array} \right).
\end{aligned}$$

Since $r(k)$ is independent of $\hat{x}(k\,|k-1\,), u(k)$ and

$$\begin{aligned}
&\mathcal{E}\left(x(k)\right) = \hat{x}(k\,|k-1\,), \mathcal{E}\left(r(k)\right) = 0, \\
&\mathcal{E}\left(x^T(k)Q_x x(k)\right) = \mathcal{E}x^T(k)Q_x\mathcal{E}x(k) \\
&+ tr\left(Q_x\mathcal{E}\left(x(k) - \mathcal{E}x(k)\right)\left(x(k) - \mathcal{E}x(k)\right)^T\right) \\
&= \hat{x}^T(k\,|k-1\,)Q_x\hat{x}(k\,|k-1\,) + tr\left(Q_x Y\right), \\
&\mathcal{E}\left((Lr(k))^T PLr(k)\right) = tr\left(PL\mathcal{E}\left(r(k)r^T(k)\right)L^T\right) = tr\left(PL\Sigma_r L^T\right),
\end{aligned}$$

it yields

$$\begin{aligned}
\min_{u(i)} J_R(i) &= \min_{u(i)} \left(\begin{bmatrix} \hat{x}^T(i\,|i-1\,) \; u^T(i) \end{bmatrix} Q \begin{bmatrix} \hat{x}(i\,|i-1\,) \\ u(i) \end{bmatrix} \right) \\
&\quad + tr\left(Q_x Y\right) + \gamma tr\left(PL\Sigma_r L^T\right) + \gamma c, \\
Q &= \begin{bmatrix} Q_x + \gamma A^T PA & Q_{xu} + \gamma A^T PB \\ Q_{ux} + \gamma B^T PA & Q_u + \gamma B^T PB \end{bmatrix}.
\end{aligned}$$

Note that the minimisation in the first term of the above equation is an LQ control problem. As a result, we have the final solution as follows:

$$\begin{aligned}
\min_{u(i)} J_R(i) &= \hat{x}^T(i\,|i-1\,)P\hat{x}(i\,|i-1\,) + c, \\
u^*(i) &= \arg\min_{u(i)} J_R(i) = K\hat{x}(i\,|i-1\,), \\
K &= -\left(Q_u + \gamma B^T PB\right)^{-1}\left(Q_{ux} + \gamma B^T PA\right),
\end{aligned}$$

with

$$P = \gamma A^T PA + Q_x - K^T \left(Q_u + \gamma B^T PB\right) K, P > 0,$$

$$c = \gamma c + tr\left(Q_x Y\right) + \gamma tr\left(PL\Sigma_r L^T\right) \Longrightarrow c = \frac{tr\left(Q_x Y\right) + \gamma tr\left(PL\Sigma_r L^T\right)}{1-\gamma}.$$

Assessment and Monitoring

On the assumption of steady state operation, it holds

$$\Delta J_{ref}(i) = \min_u J_R(i) - \min_{u_{ideal}} J(i) = \frac{tr\left(Q_x Y\right) + \gamma tr\left(PL\Sigma_r L^T\right)}{1-\gamma}. \quad (19.31)$$

ΔJ_{ref} represents the performance degradation caused by (i) the use of an observer (Kalman filter) for the state estimation, and (ii) process and measurement noises. Considering that the existence of process and measurement noises is the nature of any industrial process and a Kalman filter delivers the minimum covariance matrices of the state estimation error $(x(k) - \hat{x}(k\,|k-1))$ and residual signal $(r(k))$, ΔJ_{ref} is the minimum performance degradation. In other words, model uncertainties and unknown inputs, as described in the last section (as given in (19.13), (19.14) as well as (19.19)), may cause

$$J(i) = \hat{x}^T(i\,|i-1)P\hat{x}(i\,|i-1) + \frac{tr\left(Q_x \hat{Y}(i)\right) + \gamma tr\left(PL\hat{\Sigma}_r(i)L^T\right)}{1-\gamma}$$

becoming considerably large during real operations, so that

$$\Delta J(i) = J(i) - x_{ideal}^T(i)Px_{ideal}(i) >> \frac{tr\left(Q_x Y\right) + \gamma tr\left(PL\Sigma_r L^T\right)}{1-\gamma}. \quad (19.32)$$

In (19.32), $\hat{Y}(i)$, $\hat{\Sigma}_r(i)$ are the estimates for

$$\mathcal{E}\left(x(i) - \hat{x}(i\,|i-1)\right)\left(x(i) - \hat{x}(i\,|i-1)\right)^T, \mathcal{E}\left(r(i)r^T(i)\right),$$

respectively. This observation motivates us to introduce the following definition.

Definition 19.2 *Given $\Delta J_{ref}(i)$ and $\Delta J(i)$ defined in (19.31) and (19.32) respectively, the value*

$$P_{CPD}(i) = 1 - \frac{\Delta J_{ref}(i)}{\Delta J(i)} \quad (19.33)$$

is called the degree of control performance degradation (DCPD).

It is clear that in general

$$0 \le P_{CPD}(i) < 1,$$

and a larger P_{CPD}-value corresponds to a higher degree of performance degradation. In fact, $P_{CPD}(i)$ can also be applied for the purpose of performance-based fault detection. Given a threshold $J_{th,CPD}$ (>0) that represents the tolerant limit to the performance degradation caused by changes in the control system under consideration, an alarm is released when

$$P_{CPD}(i) > J_{th,CPD}.$$

19.2.3 Loop Performance Degradation Assessment and Monitoring

Recall that for the computation of $\Delta J(i)$, $\hat{Y}(i)$ is needed, which is generally not available. Alternatively, the loop variable e_{LPD} and LPDM (19.23) can be used for assessing and monitoring loop performance degradation. In this sub-section, we propose an alternative control performance assessment and monitoring scheme based on the loop variable e_{LPD}.

Reference System

Suppose that the ideal (nominal) system is modelled by (19.20)–(19.21) and the feedback control gain matrix F in control law (19.22) is given. Again, we adopt the system model (19.25)–(19.26) as the reference system and apply the Kalman filter (19.28)–(19.29) for the residual generation and state estimation purpose. The corresponding control law is

$$u(k) = F\hat{x}(k\,|k-1) + v(k).$$

The state space representation of LPDM (19.23) is

$$\begin{aligned} e_x(k+1) &= (A+BF)\,e_x(k) - Lr(k), \qquad (19.34)\\ e_x(k) &= x_{ideal}(k) - \hat{x}(k\,|k-1), \\ e_u(k) &= Fe_x(k), e_y(k) = (C+DF)\,e_x(k) - r(k). \qquad (19.35) \end{aligned}$$

We now introduce the following index as a reference for loop performance degradation:

$$J_{LPD,R}(i) = \mathcal{E}\sum_{k=i}^{\infty}\gamma^{k-i}\left(e_y^T(k)Q_y e_y(k) + e_u^T(k)Q_u e_u(k)\right), \qquad (19.36)$$
$$Q_y \geq 0,\, Q_u \geq 0, 0 < \gamma < 1.$$

Write $J_{LPD,R}(i)$ into

$$J_{LPD,R}(i) = \mathcal{E}\left(\left(e_y^T(i)Q_y e_y(i) + e_u^T(i)Q_u e_u(i)\right) + \gamma J_{LPD,R}(i+1)\right).$$

Note that

$$\begin{aligned}
\mathcal{E}\left(e_y^T(i)Q_y e_y(i) + e_u^T(i)Q_u e_u(i)\right) &= \mathcal{E}e_x^T(i)Qe_x(i) + \mathcal{E}r^T(i)r(i) \\
&= tr\left(Qcov\left(e_x(i)\right) + \Sigma_r\right), \\
Q &= (C+DF)^T Q_y (C+DF) + F^T Q_u F, \\
cov\left(e_x(i)\right) &= \mathcal{E}\left(x_{ideal}(i) - \hat{x}(i\,|i-1)\right)\left(x_{ideal}(i) - \hat{x}(i\,|i-1)\right)^T,
\end{aligned}$$

and recall

$$cov\left(e_x(i+1)\right) = (A+BF)\,cov\left(e_x(i)\right)(A+BF)^T + L\Sigma_r L^T$$

with $(A+BF)$ being a Schur matrix, which yields

$$\begin{aligned}
\lim_{i\to\infty} cov\left(e_x(i)\right) &= \Sigma_{e_x} > 0, \\
\Sigma_{e_x} &= (A+BF)\,\Sigma_{e_x}\,(A+BF)^T + L\Sigma_r L^T.
\end{aligned}$$

As a result, it holds in the steady state,

$$J_{LPD,R}(i) = tr\left(Q\Sigma_{e_x} + \Sigma_r\right) + \gamma J_{LPD,R}(i+1).$$

Let

$$J_{LPD,R}(i) = c.$$

We finally have

$$c = \frac{tr\left(Q\Sigma_{e_x} + \Sigma_r\right)}{1-\gamma} \Longrightarrow J_{LPD,R}(i) = \frac{tr\left(Q\Sigma_{e_x} + \Sigma_r\right)}{1-\gamma}. \tag{19.37}$$

Assessment and Monitoring

Now, we are in the position to introduce the assessment and monitoring scheme for performance degradation. Assume that the dynamics of the real control system is described by the observer-based closed-loop model (19.9) with model uncertainties or unknown inputs described by (19.13) or/and (19.14) or/and (19.19). Using (online) measurement data, state observer and residual generator as well as ideal (nominal) system model (19.20), $r(k), \hat{x}(k), x_{ideal}(k), k = i, i+1, \cdots$, can be computed online and further used for estimating Σ_{e_x}, Σ_r,

$$\hat{\Sigma}_{e_x} = \frac{1}{N}\sum_{k=i}^{N+i}\left(x_{ideal}(k) - \hat{x}(k)\right)\left(x_{ideal}(k) - \hat{x}(k)\right)^T,$$
$$\hat{\Sigma}_r = \frac{1}{N}\sum_{k=i}^{N+i} r(k)r^T(k).$$

An estimate for the (online) performance degradation is then given by

$$J_{LPD}(i) = \frac{tr\left(Q\hat{\Sigma}_{e_x} + \hat{\Sigma}_r\right)}{1-\gamma}. \tag{19.38}$$

With the same arguments for the definition of DCPD, we now introduce the concept of degree of loop performance degradation.

Definition 19.3 *Given $J_{LPD,R}(i)$ and $J_{LPD}(i)$ defined in (19.37) and (19.38) respectively, the value*

$$P_{LPD}(i) = 1 - \frac{J_{LPD,R}(i)}{J_{LPD}(i)} \tag{19.39}$$

is called the degree of loop performance degradation (DLPD).

DLPD can also be used for the fault detection purpose. It measures, to some degree and generally speaking, the difference between the ideal process input and output values and the real operating ones. If the DLPD-value is larger than a given threshold $J_{th,LPD}$,

$$P_{LPD}(i) > J_{th,LPD},$$

an alarm will be released. This indicates the loop performance degradation caused by changes in the control system cannot be accepted.

19.3 SIR of Feedback Controller and System Performances

The transfer matrix,

$$\begin{bmatrix} \hat{Y}(z) + M(z)Q(z) \\ -\hat{X}(z) + N(z)Q(z) \end{bmatrix} = \begin{bmatrix} \hat{Y}(z) + M(z)Q(z) \\ -\left(\hat{X}(z) - N(z)Q(z)\right) \end{bmatrix},$$

is the SIR of the feedback controller,

$$u(z) = K(z)y(z),\; K(z) = -\left(\hat{Y}(z) + M(z)Q(z)\right)\left(\hat{X}(z) - N(z)Q(z)\right)^{-1}.$$

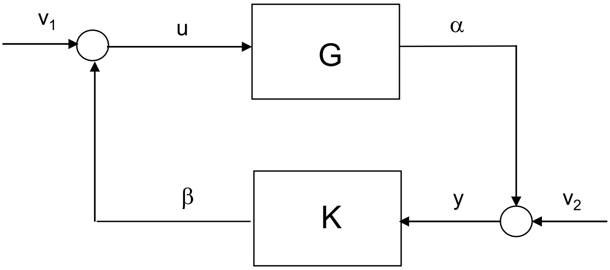


Fig. 19.2 Feedback control loop

In the rich collection of the existing methods for the design of (LTI) feedback controllers, very few methods could be identified as being dedicated to the controller design in the context of the controller SIR. In this section, we summarise some important relations between a norm of the controller SIR and different system (control) performances, which are useful for online optimisation and reconfiguration of feedback controllers in the fault-tolerant control framework.

19.3.1 Stability Margin

Stability margin is an essential control performance that, roughly speaking, indicates the stability reserve of a feedback control loop. In other words, stability margin measures the tolerant degree of a feedback controller to loop uncertainties in the context of system stability. Consider the standard feedback loop sketched in Fig. 19.2. The loop dynamics is governed by

$$\begin{aligned}\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} &= \begin{bmatrix} I & -K(z) \\ -G(z) & I \end{bmatrix}^{-1} \begin{bmatrix} v_1(z) \\ v_2(z) \end{bmatrix} \\ &= \begin{bmatrix} (I-K(z)G(z))^{-1} & (I-K(z)G(z))^{-1}K(z) \\ (I-G(z)K(z))^{-1}G(z) & (I-G(z)K(z))^{-1} \end{bmatrix} \begin{bmatrix} v_1(z) \\ v_2(z) \end{bmatrix} \\ &= \begin{bmatrix} (I-K(z)G(z))^{-1} & (I-K(z)G(z))^{-1}K(z) \\ G(z)(I-K(z)G(z))^{-1} & I+G(z)(I-K(z)G(z))^{-1}K(z) \end{bmatrix} \begin{bmatrix} v_1(z) \\ v_2(z) \end{bmatrix} \\ &= \begin{bmatrix} I+K(z)(I-G(z)K(z))^{-1}G(z) & K(z)(I-G(z)K(z))^{-1} \\ (I-G(z)K(z))^{-1}G(z) & (I-G(z)K(z))^{-1} \end{bmatrix} \begin{bmatrix} v_1(z) \\ v_2(z) \end{bmatrix}. \end{aligned}$$

In the literature, there are different concepts for introducing the definition of stability margin. For instance,

$$b_{RCF} = \left\| \begin{matrix} (I-K(z)G(z))^{-1} & (I-K(z)G(z))^{-1}K(z) \\ G(z)(I-K(z)G(z))^{-1} & G(z)(I-K(z)G(z))^{-1}K(z) \end{matrix} \right\|_{\infty}^{-1} \tag{19.40}$$

or its dual form

$$b_{LCF} = \left\| \begin{matrix} K(z)\left(I - G(z)K(z)\right)^{-1} G(z) & K(z)\left(I - G(z)K(z)\right)^{-1} \\ \left(I - G(z)K(z)\right)^{-1} G(z) & \left(I - G(z)K(z)\right)^{-1} \end{matrix} \right\|_{\infty}^{-1} \tag{19.41}$$

is a widely adopted definition. The transfer matrix in (19.40) is the one from (v_1, v_2) to (u, α), while the one in (19.41) describes the dynamics from (v_1, v_2) to (β, y), as shown in Fig. 19.2. Equivalently, we have the following relations

$$\begin{aligned} &\begin{bmatrix} \left(I - K(z)G(z)\right)^{-1} & \left(I - K(z)G(z)\right)^{-1} K(z) \\ G(z)\left(I - K(z)G(z)\right)^{-1} & G(z)\left(I - K(z)G(z)\right)^{-1} K(z) \end{bmatrix} \\ &= \begin{bmatrix} \left(I - K(z)G(z)\right)^{-1} & \left(I - K(z)G(z)\right)^{-1} K(z) \\ G(z)\left(I - K(z)G(z)\right)^{-1} & I + G(z)\left(I - K(z)G(z)\right)^{-1} K(z) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, \\ &\begin{bmatrix} K(z)\left(I - G(z)K(z)\right)^{-1} G(z) & K(z)\left(I - G(z)K(z)\right)^{-1} \\ \left(I - G(z)K(z)\right)^{-1} G(z) & \left(I - G(z)K(z)\right)^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I + K(z)\left(I - G(z)K(z)\right)^{-1} G(z) & K(z)\left(I - G(z)K(z)\right)^{-1} \\ \left(I - G(z)K(z)\right)^{-1} G(z) & \left(I - G(z)K(z)\right)^{-1} \end{bmatrix} - \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

The following results are well-known in the literature for the computation of stability margin defined in (19.40) or (19.41)

$$b_{RCF} = \left\| \begin{matrix} X_o(z) - Q(z)\hat{N}(z) & Y_o(z) + Q(z)\hat{M}(z) \end{matrix} \right\|_{\infty}^{-1}, \tag{19.42}$$

$$b_{LCF} = \left\| \begin{matrix} \hat{X}_o(z) - N(z)Q(z) \\ \hat{Y}_o(z) + M(z)Q(z) \end{matrix} \right\|_{\infty}^{-1}, \tag{19.43}$$

where $(X_o(z), Y_o(z))$, $\left(\hat{X}_o(z), \hat{Y}_o(z)\right)$ satisfy Bezout identity corresponding to the normalised RC and LC pairs of $G(z)$. In order to well understand this issue and considering that some of the results will be useful for our subsequent work, we are going to demonstrate (19.42) schematically. Equation (19.43) is the dual result of (19.42).

Since

$$\begin{aligned} &\begin{bmatrix} \left(I - K(z)G(z)\right)^{-1} & \left(I - K(z)G(z)\right)^{-1} K(z) \\ G(z)\left(I - K(z)G(z)\right)^{-1} & G(z)\left(I - K(z)G(z)\right)^{-1} K(z) \end{bmatrix} \\ &= \begin{bmatrix} I \\ G(z) \end{bmatrix} \left(I - K(z)G(z)\right)^{-1} \begin{bmatrix} I & K(z) \end{bmatrix}, \\ &G(z) = N(z) M^{-1}(z), \\ &K(z) = -\left(X(z) - Q(z)\hat{N}(z)\right)^{-1} \left(Y(z) + Q(z)\hat{M}(z)\right), \end{aligned}$$

it turns out by Bezout identity

$$\begin{aligned}&\begin{bmatrix} I \\ G(z) \end{bmatrix} (I - K(z)G(z))^{-1} \begin{bmatrix} I & K(z) \end{bmatrix}\\ &= \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \begin{bmatrix} X(z) - Q(z)\hat{N}(z) & -Y(z) - Q(z)\hat{M}(z) \end{bmatrix}.\end{aligned}$$

Let $(M_o(z), N_o(z))$ be the normalised RC pair of $G(z)$. Recall the result

$$\begin{aligned}&\begin{bmatrix} M_o(z) \\ N_o(z) \end{bmatrix} = \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} Q_o(z), \ Q_o(z) = I + (F_o - F) \left(zI - A_{F_o}\right)^{-1} B,\\ &A_{F_o} = A + BF_o,\end{aligned}$$

where F_o, F are the state feedback gain adopted in (M_o, N_o) and (M, N), respectively. It holds

$$\begin{aligned}&\left\| \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \begin{bmatrix} X(z) - Q(z)\hat{N}(z) & -Y(z) - Q(z)\hat{M}(z) \end{bmatrix} \right\|_\infty\\ &= \left\| \begin{bmatrix} M_o(z) \\ N_o(z) \end{bmatrix} Q_o^{-1}(z) \begin{bmatrix} X(z) - Q(z)\hat{N}(z) & -Y(z) - Q(z)\hat{M}(z) \end{bmatrix} \right\|_\infty\\ &= \left\| Q_o^{-1}(z) \begin{bmatrix} X(z) - Q(z)\hat{N}(z) & -Y(z) - Q(z)\hat{M}(z) \end{bmatrix} \right\|_\infty,\\ &Q_o^{-1}(z) = I + (F - F_o)(zI - A_F)^{-1} B, A_F = A + BF.\end{aligned}$$

The following lemma provides us with a useful relation between

$$\begin{aligned}&Q_o^{-1}(z) \begin{bmatrix} X(z) - Q(z)\hat{N}(z) & -Y(z) - Q(z)\hat{M}(z) \end{bmatrix} \text{ and}\\ &\begin{bmatrix} X_o(z) - Q(z)\hat{N}_o(z) & Y_o(z) + Q(z)\hat{M}_o(z) \end{bmatrix}.\end{aligned}$$

Lemma 19.1 *Given $Q_o^{-1}(z), X(z), Y(z)$, it holds*

$$\begin{aligned}&Q_o^{-1}(z)X(z) = X_o(z) - Q_F(z)\hat{N}(z), Q_o^{-1}(z)Y(z) = Y_o(z) + Q_F(z)\hat{M}(z),\\ &Q_F(z) = -(F - F_o)(zI - A_F)^{-1} L \in \mathcal{RH}_\infty,\\ &X_o(z) = (A - LC, -(B - LD), F_o, I), Y_o(z) = (A - LC, -L, F_o, 0).\end{aligned}$$

Proof Since

$$\begin{aligned}Q_o^{-1}(z)F &= F + (F - F_o)(zI - A_F)^{-1} BF\\ &= F_o + (F - F_o) + (F - F_o)(zI - A_F)^{-1} BF\\ &= F_o + (F - F_o)(zI - A_F)^{-1}(zI - A),\end{aligned}$$

it turns out

$$Q_o^{-1}(z)Y(z) = Y_o(z) - (F - F_o)(zI - A_F)^{-1}(zI - A)(zI - A_L)^{-1} L.$$

The relation

$$\begin{aligned} &(zI - A)(zI - A_L)^{-1} L = \left(I + LC(zI - A)^{-1}\right)^{-1} L \\ &= L\left(I - C(zI - A + LC)^{-1} L\right) = L\hat{M}(z) \end{aligned}$$

results in

$$Q_o^{-1}(z)Y(z) = Y_o(z) + Q_F(z)\hat{M}(z).$$

Now, consider $Q_o^{-1}(z)X(z)$, which can be written as

$$Q_o^{-1}(z)X(z) = Q_o^{-1}(z) - Q_o^{-1}(z)F(zI - A_L)^{-1}(B - LD).$$

It holds, analogue to the above study,

$$\begin{aligned} &Q_o^{-1}(z)F(zI - A_L)^{-1}(B - LD) = F_o(zI - A_L)^{-1}(B - LD) \\ &-(F - F_o)(zI - A_F)^{-1}\left(I - LC(zI - A + LC)^{-1}\right)(B - LD), \end{aligned}$$

which yields

$$\begin{aligned} &Q_o^{-1}(z)X(z) = X_o(z) + (F - F_o)(zI - A_F)^{-1} B \\ &-(F - F_o)(zI - A_F)^{-1}\left(I - LC(zI - A + LC)^{-1}\right)(B - LD) \\ &= X_o(z) + (F - F_o)(zI - A_F)^{-1} L\left(D + C(zI - A + LC)^{-1}(B - LD)\right) \\ &= X_o(z) - Q_F(z)\hat{N}(z). \end{aligned}$$

The lemma is proved.

It follows from Lemma 19.1 that

$$\begin{aligned} &Q_o^{-1}(z)\left[\, X(z) - Q(z)\hat{N}(z) \quad -Y(z) - Q(z)\hat{M}(z) \,\right] \\ &= \left[\, X_o(z) - \bar{Q}(z)\hat{N}(z) \quad -Y_o(z) - \bar{Q}(z)\hat{M}(z) \,\right], \\ &\bar{Q}(z) = Q_o^{-1}(z)Q(z) + Q_F(z) \in \mathcal{RH}_\infty. \end{aligned}$$

Moreover,

$$\begin{aligned} &\left\| \, X_o(z) - \bar{Q}(z)\hat{N}(z) \quad -Y_o(z) - \bar{Q}(z)\hat{M}(z) \, \right\|_\infty \\ &= \left\| \, X_o(z) - \bar{Q}(z)\hat{N}(z) \;\; Y_o(z) + \bar{Q}(z)\hat{M}(z) \, \right\|_\infty, \end{aligned}$$

which finally leads to the computation formula (19.42).

Remark 19.2 *It is well-known that the RCF of a transfer matrix G is not unique and depends on state feedback gain matrix F. As the dual result of Lemma 4.1, there exists the relation*

$$\begin{bmatrix} M_2(z) \\ N_2(z) \end{bmatrix} = \begin{bmatrix} M_1(z) \\ N_1(z) \end{bmatrix} Q_{21}(z),\ Q_{21}(z) = I + (F_2 - F_1)(zI - A - BF_2)^{-1} B,$$
$$M_i(z) = (A + BF_i, B, F_i, I),\ N_i(z) = (A + BF_i, B, C + DF_i, D),\ i = 1, 2.$$

According Bezout identity, there exist $(X_i(z), Y_i(z))$, $i = 1, 2$, *so that*

$$X_i(z)M_i(z) + Y_i(z)N_i(z) = I.$$

Now, multiplying $Q_{21}(z)$ *to the right side of*

$$X_1(z)M_1(z) + Y_1(z)N_1(z) = I$$

yields

$$X_1(z)M_2(z) + Y_1(z)N_2(z) = Q_{21}(z),$$

which shows that

$$Q_{12}(z)\begin{bmatrix} X_1(z) & Y_1(z) \end{bmatrix},\ Q_{12}(z) = Q_{21}^{-1}(z) \in \mathcal{RH}_\infty$$

should be a left inverse of

$$\begin{bmatrix} M_2(z) \\ N_2(z) \end{bmatrix},$$

like $\begin{bmatrix} X_2(z) & Y_2(z) \end{bmatrix}$. *Lemma 19.1 and its proof verify this result, which can be formulated in a more general form*

$$Q_{12}(z)\begin{bmatrix} X_1(z) & Y_1(z) \end{bmatrix} = \begin{bmatrix} X_2(z) & Y_2(z) \end{bmatrix} + Q(z)\begin{bmatrix} -\hat{N}_2(z) & \hat{M}_2(z) \end{bmatrix}$$
$$\Longrightarrow Q_{12}(z)\begin{bmatrix} X_1(z) & Y_1(z) \end{bmatrix}\begin{bmatrix} M_2(z) \\ N_2(z) \end{bmatrix} = I,$$
$$Q(z) = -(F_1 - F_2)(zI - A - BF_1)^{-1} L \in \mathcal{RH}_\infty.$$

A further definition of stability margin known in the literature is given by

$$b_{LCF} = \left\| \begin{bmatrix} K(z) \\ I \end{bmatrix} (I - G(z)K(z))^{-1} \hat{M}^{-1}(z) \right\|_\infty^{-1}. \tag{19.44}$$

Note that

$$\begin{bmatrix} K(z) \\ I \end{bmatrix} (I - G(z)K(z))^{-1} \hat{M}^{-1}(z) = \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix}$$

is the transfer matrix from the residual vector r to the process input and output vectors (u, y). Thus,

$$\begin{aligned} b_{LCF} &= \left\| \begin{bmatrix} K(z) \\ I \end{bmatrix} (I - G(z)K(z))^{-1} \hat{M}^{-1}(z) \right\|_{\infty}^{-1} \\ &= \left\| \begin{matrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{matrix} \right\|_{\infty}^{-1} = \left\| \begin{matrix} \hat{Y}(z) + M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{matrix} \right\|_{\infty}^{-1}. \end{aligned} \tag{19.45}$$

Below, we introduce the concept of stability margin on the basis of our closed-loop model (19.9). Without loss of generality, suppose that the uncertainty under consideration is modelled in form of the left coprime factor

$$\begin{bmatrix} \Delta_{\hat{M}} & \Delta_{\hat{N}} \end{bmatrix} \in \mathcal{H}_{\infty}.$$

It holds

$$\begin{aligned} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} &= \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} r(z) + \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} v(z) \Longrightarrow \\ \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} &= \left(I - \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} \begin{bmatrix} \Delta_{\hat{N}} & -\Delta_{\hat{M}} \end{bmatrix} \right)^{-1} \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} v(z). \end{aligned}$$

It follows from the small gain theorem that the closed-loop dynamics is stable for all $\left(\Delta_{\hat{M}}, \Delta_{\hat{N}}\right)$ if and only if

$$\left\| \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} \begin{bmatrix} \Delta_{\hat{N}} & -\Delta_{\hat{M}} \end{bmatrix} \right\|_{\infty} < 1,$$

which is equivalent to the conclusion that the closed-loop dynamics is stable for all $\left(\Delta_{\hat{M}}, \Delta_{\hat{N}}\right)$ if and only if

$$\left\| \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} \right\|_{\infty}^{-1} > \left\| \begin{bmatrix} \Delta_{\hat{N}} & -\Delta_{\hat{M}} \end{bmatrix} \right\|_{\infty}.$$

In this context, we introduce the concept of stability margin.

Definition 19.4 *Given feedback control loop model (19.9), then*

$$b_{LCF} = \left\| \begin{matrix} \hat{Y}(z) + M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{matrix} \right\|_{\infty}^{-1} \tag{19.46}$$

is called loop stability margin (LSM).

It is evident that LSM defined in (19.46) is consistent with the definitions given in (19.43) and (19.44). The dual form of b_{LCF} is

$$b_{RCF} = \left\| X(z) - Q(z)\hat{N}(z) \quad Y(z) + Q(z)\hat{M}(z) \right\|_{\infty}^{-1}. \tag{19.47}$$

Both b_{LCF} and b_{RCF} will be applied in our fault-tolerant control and performance degradation recovery framework.

Remark 19.3 *The sub-indices of b_{LCF} and b_{RCF} stand for left and right coprime factor uncertainties.*

In order to reach the maximum LSM value, we can solve the following optimisation problem

$$b_{opt}^{-1} = \min_{Q(z)\in\mathcal{RH}_\infty} b_{LCF}^{-1} = \min_{Q(z)\in\mathcal{RH}_\infty} \left\| \begin{array}{c} \hat{Y}(z) + M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{array} \right\|_\infty . \tag{19.48}$$

Theorem 19.1 *Given $(M(z), N(z))$ and $\left(\hat{X}(z), \hat{Y}(z)\right)$ as defined in (19.10)–(19.11), and let $(M_o(z), N_o(z))$ be normalised $(M(z), N(z))$ with the corresponding matrix pair $\left(\hat{X}_o(z), \hat{Y}_o(z)\right)$ satisfying*

$$\hat{X}_o(z) = (A + BF_o, L, C + DF_o, I), \hat{Y}_o(z) = (A + BF_o, -L, F_o, 0).$$

Then,

$$b_{opt}^{-1} = \min_{Q(z)\in\mathcal{RH}_\infty} \left\| \begin{array}{c} \hat{Y}_o(z) + M_o(z)Q(z) \\ \hat{X}_o(z) - N_o(z)Q(z) \end{array} \right\|_\infty . \tag{19.49}$$

In order to prove this theorem, we first introduce the following known lemma. The reference is given at the end of this chapter.

Lemma 19.2 *Given*

$$\begin{aligned} &M_i(z) = (A + BF_i, B, F_i, I), N_i(z) = (A + BF_i, B, C + DF_i, D), \\ &\hat{X}_i(z) = (A + BF_i, L, C + DF_i, I), \hat{Y}_i(z) = (A + BF_i, -L, F_i, 0), i = 1, 2, \end{aligned}$$

then it holds

$$\begin{gathered} \begin{bmatrix} \hat{X}_1(z) \\ \hat{Y}_1(z) \end{bmatrix} = \begin{bmatrix} \hat{X}_2(z) \\ \hat{Y}_2(z) \end{bmatrix} + \begin{bmatrix} M_2(z) \\ -N_2(z) \end{bmatrix} \bar{Q}(z), \\ \bar{Q}(z) = Q_{12}(z)(F_1 - F_2)(zI - A - BF_2)^{-1} L, \\ Q_{12}(z) = I + (F_1 - F_2)(zI - A - BF_1)^{-1} B. \end{gathered}$$

Proof (Proof of Theorem 19.1) It follows from the relation

$$\begin{bmatrix} M_2(z) \\ N_2(z) \end{bmatrix} = \begin{bmatrix} M_1(z) \\ N_1(z) \end{bmatrix} Q_{21}(z), Q_{21}(z) = I + (F_2 - F_1)(zI - A - BF_2)^{-1} B,$$

and Lemma 19.2 that

$$\begin{bmatrix} \hat{Y}(z) + M(z)\,Q(z) \\ \hat{X}(z) - N(z)\,Q(z) \end{bmatrix}$$

can be written as

$$\begin{aligned} \begin{bmatrix} \hat{Y}(z) + M(z)\,Q(z) \\ \hat{X}(z) - N(z)\,Q(z) \end{bmatrix} &= \begin{bmatrix} \hat{Y}_o(z) + M_o(z)\,\hat{Q}(z) \\ \hat{X}_o(z) - N_o(z)\,\hat{Q}(z) \end{bmatrix}, \\ \hat{Q}(z) = Q_o(z)\big(Q(z) + (F - F_o)&(zI - A - BF_o)^{-1} L\big) \in \mathcal{RH}_\infty, \\ Q_o(z) = I + (F - F_o)&(zI - A - BF)^{-1} B. \end{aligned}$$

As a result,

$$\begin{aligned} &\min_{Q(z)\in\mathcal{RH}_\infty} \left\| \begin{matrix} \hat{Y}(z) + M(z)\,Q(z) \\ \hat{X}(z) - N(z)\,Q(z) \end{matrix} \right\|_\infty \\ = &\min_{\hat{Q}(z)\in\mathcal{RH}_\infty} \left\| \begin{bmatrix} \hat{Y}_o(z) + M_o(z)\,\hat{Q}(z) \\ \hat{X}_o(z) - N_o(z)\,\hat{Q}(z) \end{bmatrix} \right\|_\infty . \end{aligned}$$

The theorem is thus proved.

A dual form of this result,

$$\begin{aligned} &\min_{Q(z)\in\mathcal{RH}_\infty} \left\| X(z) - Q(z)\hat{N}(z) \;\; Y(z) + Q(z)\hat{M}(z) \right\|_\infty \\ = &\min_{Q(z)\in\mathcal{RH}_\infty} \left\| X_o(z) - Q(z)\hat{N}(z) \;\; Y_o(z) + Q(z)\hat{M}(z) \right\|_\infty , \end{aligned}$$

can be proved using Lemma 19.1.

19.3.2 Residual and Fault Detectability

As discussed in Chap. 9 and Sect. 19.1, an observer-based residual vector is a function of system uncertainties (including faults). Without loss of generality, we consider uncertainty

$$\begin{bmatrix} \Delta_{\hat{M}} \;\; \Delta_{\hat{N}} \end{bmatrix} \in \mathcal{H}_\infty$$

and unknown input v_2 in the feedback control loop sketched in Fig. 19.2. On the basis of model (19.5)–(19.6), we have

$$\begin{aligned}
\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} &= \begin{bmatrix} -\hat{Y}(z) - M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} r(z) + \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \bar{v}(z), \\
\bar{v}(z) &= \left(X(z) - Q(z)\hat{N}(z) \right) v_1(z), \\
r(z) &= \hat{M}(z)y(z) - \hat{N}(z)u(z) = \Delta_{\hat{N}} u(z) - \Delta_{\hat{M}} y(z) + \left(\hat{M}(z) + \Delta_{\hat{M}} \right) v_2(z),
\end{aligned}$$

which leads to

$$r(z) = \left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right)^{-1} \begin{pmatrix} \begin{bmatrix} \Delta_{\hat{N}} & -\Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \bar{v}(z) \\ + \left(\hat{M}(z) + \Delta_{\hat{M}} \right) v_2(z) \end{pmatrix}, \quad (19.50)$$

$$U(z) = \hat{Y}(z) + M(z)Q(z), \; V(z) = \hat{X}(z) - N(z)Q(z). \quad (19.51)$$

We are interested in the influence of the SIR of the controller, $\begin{bmatrix} U(z) \\ V(z) \end{bmatrix}$, on the residual vector r and on the fault detectability. To this end, we check

$$\begin{aligned}
r_{L_2} &:= \left\| \left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right)^{-1} \right\|_\infty, \\
r_- &:= \inf_\theta \sigma_{\min} \left(\left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U\left(e^{j\theta}\right) \\ V\left(e^{j\theta}\right) \end{bmatrix} \right)^{-1} \right).
\end{aligned}$$

Here, $\sigma_{\min}(\cdot)$ denotes the minimum singular value of a matrix. When l_2-norm of r is adopted for the evaluation purpose, r_{L_2} indicates the maximal l_2-gain, while r_- can be interpreted as the minimum l_2-gain. Notice the following two inequalities:

$$\begin{aligned}
&\forall \begin{bmatrix} \Delta_{\hat{M}} & \Delta_{\hat{N}} \end{bmatrix} \in \mathcal{H}_\infty, \\
&\left\| \left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right)^{-1} \right\|_\infty \leq \frac{1}{1 - \left\| \Delta_{\hat{N}} \; \Delta_{\hat{M}} \right\|_\infty \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_\infty}, \quad (19.52)
\end{aligned}$$

$$\begin{aligned}
&\inf_\theta \sigma_{\min} \left(\left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U\left(e^{j\theta}\right) \\ V\left(e^{j\theta}\right) \end{bmatrix} \right)^{-1} \right) \quad (19.53) \\
&\geq \frac{1}{1 + \left\| \Delta_{\hat{N}} \; \Delta_{\hat{M}} \right\|_\infty \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_\infty}.
\end{aligned}$$

Moreover, for some $\begin{bmatrix} \Delta_{\hat{M}} & \Delta_{\hat{N}} \end{bmatrix} \in \mathcal{H}_\infty$, it holds

$$\left\| \left(I + \left[\Delta_{\hat{N}} \ \Delta_{\hat{M}} \right] \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right)^{-1} \right\|_{\infty} = \frac{1}{1 - \left\| \Delta_{\hat{N}} \ \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}},$$

$$\inf_{\theta} \sigma_{\min} \left(\left(I + \left[\Delta_{\hat{N}} \ \Delta_{\hat{M}} \right] \begin{bmatrix} U\left(e^{j\theta}\right) \\ V\left(e^{j\theta}\right) \end{bmatrix} \right)^{-1} \right) = \frac{1}{1 + \left\| \Delta_{\hat{N}} \ \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}}.$$

The reader is called to pay attention to the following facts:

- a robust controller will lead to a smaller r_{L_2},
- the threshold setting is proportional to r_{L_2} : a smaller r_{L_2} results in a lower threshold, and
- for a given threshold, the larger r_- is, the more sensitive r is for the changes in the loop.

In this context, we introduce the following concept.

Definition 19.5 *Given r_{L_2}, r_- defined by (19.52) and (19.53), the ratio*

$$I_{\det} := \frac{r_-}{r_{L_2}} \tag{19.54}$$

is called indicator of fault detectability in a feedback control loop.

It is evident that a larger $I_{\det}$ value indicates a higher fault detectability, and

$$\frac{1 - \left\| \Delta_{\hat{N}} \ \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}}{1 + \left\| \Delta_{\hat{N}} \ \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}} \le \frac{r_-}{r_{L_2}} \le 1. \tag{19.55}$$

Remark 19.4 *It should be emphasised that, in the context of performance-based fault detection, a fault is understood as the changes (uncertainty) in the process under consideration (the feedback control loop in this case) that are beyond the limit of technical tolerance.*

It follows from inequalities (19.52) and (19.53) that reducing $\left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}$

- enhances the system robustness against the uncertainties (including faults) in the regard of the system stability, and simultaneously results in a lower threshold, and
- increases r_-, which enhances the sensitivity of the residual to the faults (to be detected).

As a result, the lower bound of the ratio $I_{\det}$ becomes larger. In this context, it can be concluded that reducing $\left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}$ improves fault detectability. The inequality (19.55) reveals that a good fault detectability is achieved when

$$\frac{1-\left\| \Delta_{\hat{N}} \;\; \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}}{1+\left\| \Delta_{\hat{N}} \;\; \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}}$$

is (very) close to 1. In this regard, the following optimisation problem is formulated

$$\begin{aligned} &\sup_{K(z)} \frac{1-\left\| \Delta_{\hat{N}} \;\; \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}}{1+\left\| \Delta_{\hat{N}} \;\; \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U \\ V \end{bmatrix} \right\|_{\infty}} \\ \Longleftrightarrow &\min_{Q(z)\in\mathcal{RH}_{\infty}} \left\| \begin{matrix} \hat{Y}(z) + M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{matrix} \right\|_{\infty}, \end{aligned} \tag{19.56}$$

where $K(z)$ denotes stabilising controllers with (U, V) as their RC pair and being parameterised by (19.51).

19.3.3 Performance Degradation

Performance degradation issues have been intensively discussed in Sect. 19.2. In this sub-section, we continue this discussion and summarise the major results related to the RCF of the controller.

It follows from the LPDM (19.23) and the residual dynamics (19.50) that

$$\begin{aligned} e_{LPD}(z) &= \begin{bmatrix} \hat{Y}(z) + M(z)Q(z) \\ -\hat{X}(z) + N(z)Q(z) \end{bmatrix} r(z) \\ &= \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right)^{-1} \varsigma(z), \\ \varsigma(z) &= \begin{bmatrix} \Delta_{\hat{N}} & -\Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \bar{v}(z) + \left(\hat{M}(z) + \Delta_{\hat{M}} \right) v_2(z), \end{aligned}$$

and thus

$$\begin{aligned} \|e_{LPD}(z)\|_2 &\le \gamma_{LPD} \|\varsigma(z)\|_2, \\ \gamma_{LPD} &= \left\| \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \left(I + \begin{bmatrix} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{bmatrix} \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right)^{-1} \right\|_{\infty} \\ &\le \frac{\left\| \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right\|_{\infty}}{1 - \left\| \Delta_{\hat{N}} \;\; \Delta_{\hat{M}} \right\|_{\infty} \left\| \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right\|_{\infty}}. \end{aligned} \tag{19.57}$$

Definition 19.6 *Given value γ_{LPD} defined in (19.57) and assume that*

$$\left\| \begin{array}{cc} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{array} \right\|_{\infty} \leq \delta,$$

then

$$\bar{\gamma}_{LPD} := \sup_{\left\| \begin{array}{cc} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{array} \right\|_{\infty} \leq \delta} \gamma_{LPD} \tag{19.58}$$

is called loop performance degradation coefficient.

It is obvious that $\bar{\gamma}_{LPD}$ is a function of the feedback controller and parameterised by $Q(z) \in \mathcal{RH}_{\infty}$. Moreover,

$$\min_{Q(z) \in \mathcal{RH}_{\infty}} \left\| \begin{array}{c} \hat{Y}(z) + M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{array} \right\|_{\infty} = \min_{Q(z) \in \mathcal{RH}_{\infty}} \left\| \begin{bmatrix} U(z) \\ V(z) \end{bmatrix} \right\|_{\infty}$$

leads to

$$\min_{Q(z) \in \mathcal{RH}_{\infty}} \bar{\gamma}_{LPD} = \min_{Q(z) \in \mathcal{RH}_{\infty}} \sup_{\left\| \begin{array}{cc} \Delta_{\hat{N}} & \Delta_{\hat{M}} \end{array} \right\|_{\infty} \leq \delta} \gamma_{LPD} =: \gamma_{LPD}^{*}. \tag{19.59}$$

19.3.4 Summary: A Unified Perspective

In this section, we have defined and discussed three different system performances. They are

- stability margin,
- fault detectability indicator and
- loop performance degradation coefficient.

Although these three different system performances represent and reflect three different structural properties of a feedback control loop, their optimisation can be achieved unifiedly by minimising the $\mathcal{H}_{\infty}$-norm of the SIR of the adopted feedback controller. This result seems a little surprised, but can be well understood and interpreted from the following unified perspective.

Recall that the SIR of the feedback controller is in fact an observer driven by the residual signal r. Moreover, the outputs of the SIR, u and y, can be written as

$$u(z) = F\hat{x}(z) - Q(z)r(z), \, y(z) = (C + DF)\,\hat{x}(z) + (I - DQ(z))\,r(z),$$

which are the estimates for

$$Fx(k) \text{ and } (C + DF)\,x(k),$$

as described in Sect. 5.3 and according to the parameterisation of functional observers. Here, $x(k)$ is the state variables of the nominal system that is free of disturbances and uncertainties, and $\hat{x}(k)$ is its estimate. On the other hand, the residual signal r is driven by the disturbances and uncertainties which exist in the control loop. Thus, minimising the SIR of the feedback controller is equivalent to the minimisation of the transfer function from the disturbances and uncertainties to the estimates for $Fx(k)$ and $(C + DF)\,x(k)$. This is the unified perspective of our study on the three system performances, which can also be called information and estimation perspective of control and detection. In fact, this insight interpretation gives a more general form of the well-known separation principle, in which the design (and optimisation) of a feedback controller can be performed by

- the design of a state feedback gain matrix F and
- an optimal estimator for $Fx(k)$ and $(C + DF)\,x(k)$, parameterised by

$$\begin{bmatrix} \hat{Y}(z) + M(z)\,Q(z) \\ \hat{X}(z) - N(z)\,Q(z) \end{bmatrix}.$$

19.4 Notes and References

In this chapter, some essential control and detection issues have been addressed from the aspects different from those known in the well-established control theoretical framework. Our focus has been on the handling of uncertainties, which include faults, and their representation by residual signals, and on performance degradation caused by the uncertainties. Some of the results are new, and all discussions serve for our works in the subsequent chapters.

The observer-based input-output model (19.5)–(19.6) is the first novel result, which not only provides us with a new type of model forms, but also opens a new way to deal with uncertainties and faults, and thus is of essential importance for our subsequent investigation. We would like to emphasise the role of the residual signal in this model, which allows to handle uncertainties, being typically not accessible, by means of the available residual signal, as illustrated in Fig. 19.1. In fact, this model can be applied to dealing with some standard control problems alternatively, as done in Sect. 19.2 for solving an LQG-like optimisation problem. The interested reader may try, for instance, to solve $\mathcal{H}_2$-controller design problem based on this model. Indeed, this model can also be understood as a natural demonstration of the well-known separation principle.

LTR is a classic topic of control theory and its introduction can be found, for instance, in [1, 2]. The classic LTR concept deals with recovering control performance degradation caused by the use of a state estimate in an observer-based state feedback controller. In our study, we have extended the LTR concept to the assessment and monitoring of system performance degradation in a more general context and with the focus of system performance degradation caused by the uncertainties. We have introduced the vector-valued variable e_{LPD} representing the difference between the ideal and real system input and output vectors, and derived the loop performance degradation model (19.23). For the control performance assessment and monitoring, two schemes have been introduced. The first one is dedicated to the assessment of control degradation performance in real operations with respect to an ideal LQ controller. It is evident that this scheme is less practical, since not every controller would be designed in the sense of LQ-optimum. The second scheme is more applicable and provides us with a real-time loop performance assessment with respect to a given (state feedback) controller. In both schemes, the reference process is assumed to be corrupted with white process and measurement noises. Correspondingly, a Kalman filter is applied for the state estimation and residual generation. As a result, the performance degradations in real operations are measured with respect to the reference values, which indicate the performance degradations caused by the use of the Kalman filter and the existence of the white process and measurement noises. In this context, the concepts of degree of control performance degradation P_{CPD} as well as degree of loop performance degradation P_{LPD} have been introduced. In our study, some standard design and analysis methods have been applied, including dynamic programming, LQ and LQG controller design. We refer the reader to [3–5] for more details.

The last part of our work in this chapter deals with the relations between the SIR of a feedback controller and three different system performances, including stability margin, fault detectability indicator and loop performance degradation coefficient. Stability margin is a classic concept in robust control theory, although there exists no clear definition. The definition of stability margin b_{LCF} as well as its dual form b_{RCF} are commonly used in the literature, for instance, in [6–8]. In book [9] , a slightly different definition of stability margin is given. In our work, we have defined stability margin in Definition 19.4 on the basis of the loop model (19.9). Independent of the original definitions, the computation formula for the stability margin is identical and given by

$$\left\| \begin{bmatrix} \hat{Y}(z) + M(z)Q(z) \\ \hat{X}(z) - N(z)Q(z) \end{bmatrix} \right\|_{\infty}^{-1}.$$

That is the inverse of the $\mathcal{H}_\infty$-norm of the SIR of the adopted controller.

Our study on fault detectability is in fact a summary of the major results in Chap. 9 on the similar topic. We have introduced the concept of indicator of fault detectability I_{det}, which gives the ratio of the minimum influence of uncertainties on the residual to the maximum influence. Recalling the fact that threshold is set proportional to the maximum influence of the uncertainties on the residual, increasing the value of this ratio leads to improvement of fault detectability. We have demonstrated that the difference of the real I_{det} value to the ideal value (equal to one) can be reduced by minimising $\mathcal{H}_\infty$-norm of the controller SIR.

Finally, we have introduced loop performance degradation coefficient $\bar{\gamma}_{LPD}$ as a measurement of loop performance degradation, which can be, for example, used in system design work. Also $\bar{\gamma}_{LPD}$ depends on the $\mathcal{H}_\infty$-norm of the controller SIR and can be reduced by minimising it.

At the end of this work, we have given an insight understanding why minimising the $\mathcal{H}_\infty$-norm of the controller SIR results in unified optimisation of all three system performances: stability margin, fault detectability indicator and loop performance degradation coefficient. We would like to emphasise this unified perspective. The outputs of the SIR of the adopted controller, u and y, are in fact the estimates for

$$Fx(k) \text{ and } (C + DF)\,x(k),$$

the ideal state feedback gain and the corresponding process output. Minimising the SIR leads to the minimisation of the transfer function from the uncertainties to the estimates for $Fx(k)$ and $(C + DF)\,x(k)$. This is a more general form of the well-known separation principle.

References

1. T.-T. Tay, I. Mareels, and J. B. Moore, *High Performance Control*. Springer Science + Business Media, 1998.
2. K. Zhou, *Essential of Robust Control*. Englewood Cliffs, NJ: Prentice-Hall, 1998.
3. G. Chen, G. Chen, and S.-H. Hsu, *Linear Stochastic Control Systems*. CRC Press, 1995.
4. B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control: A Unified Approach to H_2 and H_{inf} Theories*. SIAM studies in applied and numerical mathematics, 1999.
5. V. Kucera, *Analysis and Design of Discrete Linear Control Systems*. Prentice Hall, 1991.
6. M. Vidyasagar and H. Kimura, "Robust controllers for uncertain linear variable systems," *Automatica*, vol. 22, pp. 85–94, 1986.
7. T. T. Georgiou and M. C. Smith, "Optimal robustness in the gap metric," *IEEE Trans. AC*, vol. 35, pp. 673–686, 1990.
8. G. Vinnicombe, *Uncertainty and Feedback: H_{inf} Loop-Shaping and the V-Gap Metric*. World Scientific, 2000.
9. S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control*. John Wiley and Sons, Ltd, 2005.

Chapter 20
Performance Recovery and Fault-Tolerant Control Schemes

20.1 Reviewing LQ Control Problems

In this section, we review the LQ control problems from the viewpoint of fault-tolerant control and recovering performance degradation. It builds the fundament for our work on fault-tolerant control, performance degradation recovery, and online observer optimisation.

20.1.1 LQG Control Problem

We consider an extended form of the nominal LTI system given in (19.1) as well as its minimal state space realisation (19.2), which is described by

$$x(k+1) = Ax(k) + Bu(k) + w(k), x(0) = x_0, \tag{20.1}$$
$$y(k) = Cx(k) + Du(k) + q(k). \tag{20.2}$$

Here, w, q are white noises satisfying (19.27). By means of the observer-based input–output model (19.5)–(19.6) with the following observer and observer-based residual generator,

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L\left(y(k) - \hat{y}(k)\right), \tag{20.3}$$
$$r(k) = y(k) - \hat{y}(k), \hat{y}(k) = C\hat{x}(k) + Du(k), \tag{20.4}$$

the system (20.1)–(20.2) is written into

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr(k), \tag{20.5}$$
$$y(k) = r(k) + C\hat{x}(k) + Du(k). \tag{20.6}$$

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_20

Note that the dynamics of the residual generator is governed by

$$\begin{aligned} e\left(k+1\right) &= \left(A-LC\right)e\left(k\right)+w(k)-Lq(k), \\ r(k) &= Ce(k)+q(k), e\left(k\right)=x(k)-\hat{x}(k) \Longleftrightarrow \\ r(z) &= C\left(zI-A_L\right)^{-1}\left(w(z)-Lq(z)\right)+q(z), A_L=A-LC. \end{aligned}$$

The (nominal) LQG control problem addressed in the subsequent work is formulated as: Given system model (20.1)–(20.2) and observer (20.3)–(20.4), find the feedback gain matrix F of the control law

$$u(k)=F\hat{x}(k)+v(k)$$

and the observer gain matrix L so that the cost function

$$\begin{aligned} J(i) &= \mathcal{E}\sum_{k=i}^{\infty}\gamma^{k-i}\left(y^T(k)Q_y y(k)+u^T(k)Q_u u(k)\right), \qquad (20.7) \\ Q_y &\geq 0, Q_u+D^TQ_yD>0, 0<\gamma<1, \end{aligned}$$

is minimised.

Remark 20.1 *It is noteworthy to mention that the expectation in the above cost function is a conditional expected value. That is, the expectation under condition of given measurement data up to the sampling time $k=i$.*

LQG problem with the cost function (20.7) is standard in control theory, and well described in many textbooks on modern control theory. This problem can be approached using various well-established techniques and the solutions are well-known. For our purpose, we will re-study the formulated LQG problem based on the observer-based input–output model (20.5)–(20.6) and focus on some aspects, which are of special interests and importance for our work on fault-tolerant control and recovering performance degradation.

20.1.2 On Solutions of LQG Control Problem

Case I: *F* is given, and *r(k)* and *$\hat{x}$(k)* are uncorrelated

We first derive the value of the cost function for a given feedback control gain F and observer gain L. For our purpose, re-write the cost functions (20.7) on the basis of the observer-based input–output model (19.5)–(19.6). It yields

$$\begin{aligned}
&y^T(k)Q_y y(k) + u^T(k)Q_u u(k) \\
&= \left(C_F\hat{x}(k) + r(k)\right)^T Q_y \left(C_F\hat{x}(k) + r(k)\right) + \left(F\hat{x}(k)\right)^T Q_u F\hat{x}(k) \\
&= \left[\hat{x}^T(k)\ r^T(k)\right] \begin{bmatrix} C_F^T Q_y C_F + F^T Q_u F & C_F^T Q_y \\ Q_y C_F & Q_y \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ r(k) \end{bmatrix}, \\
&C_F = C + DF.
\end{aligned}$$

For the sake of simplicity, $v(k)$ is set equal to zero in the above expression. We assume, at first, that

- $r(k)$ is uncorrelated with $\hat{x}(k)$,
- the observer (20.3)–(20.4) is operating in the steady state and thus

$$\begin{aligned}
&\mathcal{E}r(k) = 0, \mathcal{E}r(k)r^T(k) = \Sigma_r > 0, \\
&\mathcal{E}\left(x(k) - \hat{x}(k)\right)\left(x(k) - \hat{x}(k)\right)^T =: \Sigma_x.
\end{aligned}$$

Let

$$\begin{aligned}
J(i) &= \mathcal{E}\sum_{k=i}^{\infty}\gamma^{k-i}\left[\hat{x}^T(k)\ r^T(k)\right]\begin{bmatrix} C_F^T Q_y C_F + F^T Q_u F & C_F^T Q_y \\ Q_y C_F & Q_y \end{bmatrix}\begin{bmatrix} \hat{x}(k) \\ r(k) \end{bmatrix} \\
&= \hat{x}^T(i)P\hat{x}(i) + c.
\end{aligned} \tag{20.8}$$

By defining

$$Q_{xr} = \begin{bmatrix} C_F^T Q_y C_F + F^T Q_u F & C_F^T Q_y \\ Q_y C_F & Q_y \end{bmatrix}, A_F = A + BF,$$

write $J(i)$ into

$$\begin{aligned}
J(i) &= \mathcal{E}\left(\left[\hat{x}^T(i)\ r^T(i)\right]Q_{xr}\begin{bmatrix} \hat{x}(i) \\ r(i) \end{bmatrix} + \gamma J(i+1)\right) \\
&= \mathcal{E}\left[\hat{x}^T(i)\ r^T(i)\right]Q_{xr}\begin{bmatrix} \hat{x}(i) \\ r(i) \end{bmatrix} + \mathcal{E}\gamma\hat{x}^T(i+1)P\hat{x}(i+1) + \gamma c \\
&= \hat{x}^T(i)\left(\gamma A_F^T P A_F + C_F^T Q_y C_F + F^T Q_u F\right)\hat{x}(i) \\
&\quad + tr\left(\left(C_F^T Q_y C_F + F^T Q_u F\right)\Sigma_x\right) + tr\left(\left(Q_y + \gamma L^T P L\right)\Sigma_r\right) + \gamma c.
\end{aligned}$$

Note that in the above computation, the assumptions on $\hat{x}(i), r(k)$ are utilised. As a result, (20.8) holds if $P > 0$ solves the following Lyapunov equation

$$P = \gamma A_F^T P A_F + C_F^T Q_y C_F + F^T Q_u F,$$

and c satisfies

$$\begin{aligned} c &= tr\left(\left(C_F^T Q_y C_F + F^T Q_u F\right)\Sigma_x\right) + tr\left(\left(Q_y + \gamma L^T P L\right)\Sigma_r\right) + \gamma c \\ \Longrightarrow c &= \frac{tr\left(\left(C_F^T Q_y C_F + F^T Q_u F\right)\Sigma_x\right) + tr\left(\left(Q_y + \gamma L^T P L\right)\Sigma_r\right)}{1-\gamma}. \end{aligned}$$

Case II: Optimal Solution

Now, we would like to find the optimal feedback gain F on the above assumptions on $r(k)$ and $\hat{x}(k)$. Write

$$\begin{aligned} & y^T(k) Q_y y(k) + u^T(k) Q_u u(k) \\ &= \left[\hat{x}^T(k)\ u^T(k)\ r^T(k)\right] Q_{xur} \begin{bmatrix} \hat{x}(k) \\ u(k) \\ r(k) \end{bmatrix}, \\ Q_{xur} &= \begin{bmatrix} C^T Q_y C & C^T Q_y D & C^T Q_y \\ D^T Q_y C & Q_u + D^T Q_y D & D^T Q_y \\ Q_y C & Q_y D & Q_y \end{bmatrix}, \end{aligned}$$

and further, on the assumption of (20.8),

$$\begin{aligned} J(i) &= \mathcal{E}\left(\left[\hat{x}^T(i)\ u^T(i)\ r^T(i)\right] Q_{xur} \begin{bmatrix} \hat{x}(i) \\ u(i) \\ r(i) \end{bmatrix} + \gamma J(i+1)\right) \\ &= \mathcal{E}\left(\left[\hat{x}^T(i)\ u^T(i)\ r^T(i)\right] Q_{xur} \begin{bmatrix} \hat{x}(i) \\ u(i) \\ r(i) \end{bmatrix} + \gamma \hat{x}^T(i+1) P \hat{x}(i+1) + \gamma c\right) \\ &= \left(\begin{array}{c} \left[\hat{x}^T(i)\ u^T(i)\right] Q_{xu} \begin{bmatrix} \hat{x}(i) \\ u(i) \end{bmatrix} + tr\left(C^T Q_y C \Sigma_x\right) \\ + tr\left(\left(Q_y + \gamma L^T P L\right)\Sigma_r\right) + \gamma c \end{array}\right), \\ Q_{xu} &= \begin{bmatrix} \gamma A^T P A + C^T Q_y C & \gamma A^T P B + C^T Q_y D \\ \gamma B^T P A + D^T Q_y C & Q_u + D^T Q_y D + \gamma B^T P B \end{bmatrix}. \end{aligned}$$

It is straightforward that

$$F = -\left(Q_u + D^T Q_y D + \gamma B^T P B\right)^{-1}\left(\gamma B^T P A + D^T Q_y C\right) \tag{20.9}$$

leads to

$$J(i) = \min_F J(i) = \hat{x}^T(i) P \hat{x}(i) + c$$

with $P > 0$ solving the following Riccati equation

$$P = \gamma A^T PA + C^T Q_y C - F^T \left(Q_u + D^T Q_y D + \gamma B^T PB\right) F$$

and c given by

$$c = \frac{tr\left(C^T Q_y C \Sigma_x\right) + tr\left(\left(Q_y + \gamma L^T PL\right) \Sigma_r\right)}{1-\gamma}.$$

Recall our assumptions on $r(k)$ and $\hat{x}(k)$. $r(k)$ is uncorrelated with $\hat{x}(k)$ if and only if $r(k), r(k-1), \ldots,$ are uncorrelated. It implies that $r(k)$ should be white. Consequently, the adopted observer (20.3)–(20.4) must be a Kalman filter.

Summarising the above results makes it clear that the optimal solution to the LQG problem is given by (20.9) for F and Kalman filter gain for L, as we know from the standard solution.

Case III: F, L are given, $r(k)$ and $\hat{x}(k)$ are correlated

At the end of this study, we would like to give a general solution for the case that both the feedback gain F and observer gain L are given with L being different from the Kalman filter gain matrix. In other words, $r(k)$ and $x(k)$ are correlated. The assumption that the observer is operating in the steady state with

$$\mathcal{E}r(k) = 0, \mathcal{E}r(k)r^T(k) = \Sigma_r > 0, \mathcal{E}\left(x(k) - \hat{x}(k)\right)\left(x(k) - \hat{x}(k)\right)^T = \Sigma_x$$

still holds. It is obvious that the cost function also depends on $r(k)$. This requires, as the problem solution, to take the dynamics of the residual generator,

$$\begin{aligned} e(k+1) &= (A - LC)e(k) + w(k) - Lq(k), \\ r(k) &= Ce(k) + q(k), e(k) = x(k) - \hat{x}(k), \end{aligned}$$

into account. Consider the cost function (20.8). It turns out

$$\begin{aligned} J(i) &= \mathcal{E}\left(\left[\hat{x}^T(i)\ r^T(i)\right] Q_{xr} \begin{bmatrix} \hat{x}(i) \\ r(i) \end{bmatrix} + \gamma J(i+1)\right) \\ &= \left(\mathcal{E}\left[\hat{x}^T(i)\ r^T(i)\right] Q_{xr} \begin{bmatrix} \hat{x}(i) \\ r(i) \end{bmatrix} + \gamma \mathcal{E}\hat{x}^T(i+1)P\hat{x}(i+1) + \gamma c\right). \end{aligned}$$

According to the rule,

$$\mathcal{E}\left(\zeta^T \Psi \zeta\right) = \mathcal{E}\zeta^T \Psi \mathcal{E}\zeta + tr\left(\Sigma_\zeta \Psi\right), \Sigma_\zeta = \mathcal{E}\left((\zeta - \mathcal{E}\zeta)(\zeta - \mathcal{E}\zeta)^T\right),$$

where ζ is a random vector and Ψ a matrix of appropriate dimensions, it turns out

$$\begin{aligned}
\mathcal{E}\left[\hat{x}^T(i)\ r^T(i)\right] Q_{xr}\begin{bmatrix}\hat{x}(i)\\ r(i)\end{bmatrix} &= \mathcal{E}\left[\hat{x}^T(i)\ (Ce(i)+q(i))^T\right] Q_{xr}\begin{bmatrix}\hat{x}(i)\\ Ce(i)+q(i)\end{bmatrix}\\
&= \left[\hat{x}^T(i)\ \ 0\right] Q_{xr}\begin{bmatrix}\hat{x}(i)\\ 0\end{bmatrix}\\
&\quad + tr\left(\left[0\ \ C^T\right] Q_{xr}\begin{bmatrix}0\\ C\end{bmatrix}\Sigma_x\right) + tr\left(Q_y\Sigma_q\right)\\
&= \hat{x}^T(i)\left(C_F^T Q_y C_F + F^T Q_u F\right)\hat{x}(i) + tr\left(C^T Q_y C\Sigma_x\right)\\
&\quad + tr\left(Q_y\Sigma_q\right),\\
\mathcal{E}\hat{x}^T(i+1)P\hat{x}(i+1) &= \mathcal{E}\hat{x}^T(i+1)P\mathcal{E}\hat{x}(i+1) + tr\left(L\mathcal{E}\left(r(i)r^T(i)\right)L^T P\right)\\
&= \hat{x}^T(i)A_F^T P A_F\hat{x}(i) + tr\left(L^T P L\Sigma_r\right).
\end{aligned}$$

It yields

$$\begin{aligned}
&J(i) = \hat{x}^T(i)\bar{Q}_x\hat{x}(i) + tr\left(C^T Q_y C\Sigma_x\right) + tr\left(Q_y\Sigma_q\right) + \gamma tr\left(L^T P L\Sigma_r\right) + \gamma c,\\
&\bar{Q}_x = C_F^T Q_y C_F + F^T Q_u F + \gamma A_F^T P A_F.
\end{aligned}$$

Finally, we have

$$J(i) = \hat{x}^T(i)P\hat{x}(i) + c \tag{20.10}$$

with $P > 0$ solving

$$P = \gamma A_F^T P A_F + C_F^T Q_y C_F + F^T Q_u F, \tag{20.11}$$

and c given by

$$c = \frac{tr\left(C^T Q_y C\Sigma_x\right) + tr\left(Q_y\Sigma_q\right) + \gamma tr\left(L^T P L\Sigma_r\right)}{1-\gamma}. \tag{20.12}$$

Equations (20.10)–(20.12) demonstrate that

- controller and observer optimisation can be realised separately, and
- the value of the cost function $J(i)$ will be reduced, when Σ_x, Σ_r become weak. This can be achieved by optimising the observer.

20.1.3 On Solutions of LQR Control Problem

LQ control for systems with deterministic disturbances is often called LQR (regulator) or $\mathcal{H}_2$ control problem. In the LQR study, the disturbance under consideration is the (unknown) initial state variables, which can also be equivalently formulated as an impulse disturbance. In the framework of $\mathcal{H}_2$ control, the controller design is achieved by minimising the $\mathcal{H}_2$-norm of the transfer function from the disturbances under consideration to the control output variables. Although the control problems are handled in different ways, these two problems are equivalent, since the transfer function is the z-transform of the impulse response (of the control output to the impulse disturbance). In the following, we will focus on the LQR problem. We will also briefly illustrate the $\mathcal{H}_2$ control solution based on the observer-based input–output model (19.5)–(19.6).

Consider the observer-based input–output model (20.5)–(20.6) with the dynamics of the residual generator

$$e\left(k+1\right)=\left(A-LC\right)e\left(k\right), r(k)=Ce(k), e\left(0\right)=x(0)-\hat{x}(0)\neq 0,$$

and the cost function

$$\begin{aligned} J(i) &= \sum_{k=i}^{\infty}\gamma^{k-i}\left(y^T(k)Q_y y(k)+u^T(k)Q_u u(k)\right), \\ Q_y &\geq 0, Q_u + D^T Q_y D > 0, 0 < \gamma \leq 1. \end{aligned} \tag{20.13}$$

For given F, L, the cost function can be further written as

$$\begin{aligned} J(i) &= \sum_{k=i}^{\infty}\gamma^{k-i}\left(\begin{bmatrix}\hat{x}^T(k) & e^T(k)\end{bmatrix}Q_{xe}\begin{bmatrix}\hat{x}(k)\\ e(k)\end{bmatrix}\right), \\ Q_{xe} &= \begin{bmatrix} C_F^T Q_y C_F + F^T Q_u F & C_F^T Q_y C \\ C^T Q_y C_F & C^T Q_y C\end{bmatrix}. \end{aligned}$$

It is well-known that if

$$A_F = A + BF, A_L = A - LC$$

are Schur matrices,

$$\begin{aligned} J(i) &= \begin{bmatrix}\hat{x}^T(i) & e^T(i)\end{bmatrix}P\begin{bmatrix}\hat{x}(i)\\ e(i)\end{bmatrix}, \\ P &= Q_{xe} + \gamma\begin{bmatrix}A_F & LC\\ 0 & A_L\end{bmatrix}^T P\begin{bmatrix}A_F & LC\\ 0 & A_L\end{bmatrix} > 0. \end{aligned}$$

This can be demonstrated by the following computation,

$$\begin{aligned} J(i) &= \left[\hat{x}^T(i)\ e^T(i) \right] Q_{xe} \begin{bmatrix} \hat{x}(i) \\ e(i) \end{bmatrix} + \gamma J(i+1) \\ &= \left[\hat{x}^T(i)\ e^T(i) \right] \left(Q_{xe} + \gamma \begin{bmatrix} A_F & LC \\ 0 & A_L \end{bmatrix}^T P \begin{bmatrix} A_F & LC \\ 0 & A_L \end{bmatrix} \right) \begin{bmatrix} \hat{x}(i) \\ e(i) \end{bmatrix} \\ &= \left[\hat{x}^T(i)\ e^T(i) \right] P \begin{bmatrix} \hat{x}(i) \\ e(i) \end{bmatrix}. \end{aligned}$$

To study the coupling between $\hat{x}(i)$ and $e(i)$ and also to be consistent with the standard $\mathcal{H}_2$ control formulation, we introduce

$$\alpha(k) = C_\alpha x(k) + D_\alpha u(k) \in \mathcal{R}^m, rank\left(D_\alpha^T D_\alpha\right) = p$$

and write the cost function in terms of

$$J = \|\alpha(k)\|_2^2 = \sum_{k=0}^{\infty} \alpha^T(k)\alpha(k).$$

Moreover, it is assumed that the system model under consideration is

$$x\left(k+1\right) = Ax\left(k\right) + Bu(k) + E_d d(k), x(0) = x_0, \tag{20.14}$$

$$y(k) = Cx(k) + Du(k) + F_d d(k), \tag{20.15}$$

where

$$d(k) = \begin{bmatrix} d_1(k) \\ \vdots \\ d_{k_d}(k) \end{bmatrix} \in \mathcal{R}^{k_d}, d_i(k) = \delta(k) = \begin{cases} 1, k = 0, \\ 0, k \neq 0, \end{cases}, i = 1, \cdots k_d,$$

and $E_d \in \mathcal{R}^{n \times k_d}$, $F_d \in \mathcal{R}^{m \times k_d}$ are known matrices of appropriate dimensions with

$$rank\left(F_d F_d^T\right) = m.$$

Note that the dynamics of the residual generator is, in this case, governed by

$$\begin{aligned} e\left(k+1\right) &= A_L e\left(k\right) + E_L d(k), r(k) = Ce(k) + F_d d(k), \\ A_L &= A - LC, E_L = E_d - LF_d. \end{aligned}$$

It follows from the observer-based input–output model (20.5)–(20.6) that

$$\begin{aligned}\alpha(k) &= C_{\alpha,F}\hat{x}(k) + C_{\alpha}e(k), C_{\alpha,F} = C_{\alpha} + D_{\alpha}F \Longrightarrow \\ \alpha(z) &= C_{\alpha,F}\left(zI - A_F\right)^{-1} Lr(z) + C_{\alpha}e(z) \\ &= C_{\alpha,F}\left(zI - A_F\right)^{-1} LG_{rd}(z) + C_{\alpha}G_{ed}(z), \\ G_{rd}(z) &= C\left(zI - A_L\right)^{-1} E_L + F_d, G_{ed}(z) = \left(zI - A_L\right)^{-1} E_L.\end{aligned}$$

Denote the response of $\alpha(k)$ to $d_i(k)$ by $\alpha_i(k)$. Since $\delta(k)$ is the unit impulse, it holds

$$J = \sum_{i=1}^{k_d} \|\alpha_i(k)\|_2^2 = \left\|C_{\alpha,F}\left(zI - A_F\right)^{-1} LG_{rd}(z) + C_{\alpha}G_{ed}(z)\right\|_2^2. \tag{20.16}$$

We would like to call the reader's attention that the norm on the left hand side of the above equation is the sum of the l_2-norm of signals $\alpha_i(k), i = 1, \cdots, k_d$, and the norm on the right hand side is the $\mathcal{H}_2$-norm of the transfer function from the disturbances to $\alpha(k)$. Now, let the observer-based residual generator (20.3)–(20.4) be the so-called unified FDF described in Sect. 4.3, which is also an $\mathcal{H}_2$-observer. We know

$$G_{ed}(z)G_{ed}^T(z^{-1}) = CXC^T + F_dF_d^T$$

with $X > 0$ solving the Riccati equation

$$AXA^T + E_dE_d^T - L\left(CXC^T + F_dF_d^T\right)L^T = X > 0, \tag{20.17}$$

$$L = \left(AXC^T + E_dF_d^T\right)\left(CXC^T + F_dF_d^T\right)^{-1}. \tag{20.18}$$

As a result, it holds

$$\begin{aligned}&\left\|C_{\alpha,F}\left(zI - A_F\right)^{-1} LG_{rd}(z) + C_{\alpha}G_{ed}(z)\right\|_2^2 \\ &= \left\|C_{\alpha,F}\left(zI - A_F\right)^{-1} L\left(CXC^T + F_dF_d^T\right)^{1/2} + C_{\alpha}G_{ed}(z)U^T(z^{-1})\right\|_2^2, \\ &U(z) = \left(CXC^T + F_dF_d^T\right)^{-1/2} G_{ed}(z) \Longrightarrow U(z)U^T(z^{-1}) = I.\end{aligned}$$

It is straightforward to check

$$C_{\alpha}G_{ed}(z)U^T(z^{-1}) \in \mathcal{RH}_2^{\perp},$$

and moreover

$$\begin{aligned}&C_{\alpha,F}\left(zI - A_F\right)^{-1} L\left(CXC^T + F_dF_d^T\right)^{1/2} \in \mathcal{RH}_2, \\ &\left\|C_{\alpha}G_{ed}(z)U^T(z^{-1})\right\|_2^2 = \|C_{\alpha}G_{ed}(z)\|_2^2.\end{aligned}$$

Thus, we have

$$J = \|\alpha(k)\|_2^2 = \left\| C_{\alpha,F} (zI - A_F)^{-1} \bar{L} \right\|_2^2 + \|C_\alpha G_{ed}(z)\|_2^2 , \tag{20.19}$$
$$\bar{L} = L \left(CXC^T + F_d F_d^T \right)^{1/2} = \left(AXC^T + E_d F_d^T \right) \left(CXC^T + F_d F_d^T \right)^{-1/2} .$$

Equation (20.19) reveals that

- once the observer gain matrix L is set according to (20.18), which gives a unified FDF and an $\mathcal{H}_2$-observer, the influence of the residual signal $r(k)$ and the state estimation error $e(k)$ on the cost function J is decoupled,
- from the control and estimation point of view, tuning feedback control gain matrix F can reduce the first term in the cost function,

$$\left\| C_{\alpha,F} (zI - A_F)^{-1} \bar{L} \right\|_2^2 ,$$

  while optimising the observer gain matrix L will lead to reduction in the second term $\|C_\alpha G_{ed}(z)\|_2^2$, and
- this allows a separate and parallel optimisation of the controller and observer towards an overall optimisation of the control loop.

It is evident that

$$F = -\left(D_\alpha^T D_\alpha + B^T P B \right)^{-1} \left(D_\alpha^T C_\alpha + B^T P A \right) ,$$
$$P = A^T P A + C_\alpha^T C_\alpha - F^T \left(D_\alpha^T D_\alpha + B^T P B \right) F > 0$$

is the solution of the optimisation problem

$$\min_F \left\| C_{\alpha,F} (zI - A_F)^{-1} \bar{L} \right\|_2^2 ,$$

which is equivalent to the LQ control problem

$$\min_{u(k)} \|\alpha(k)\|_2^2$$
$$\text{s.t. } x(k+1) = Ax(k) + Bu(k), x(0) = x_0 .$$

In summary, the optimal LQR problem consists of the optimal solution of LQ (state feedback) control and an $\mathcal{H}_2$-optimal observer (the unified FDF), as we expect and know from the standard solution.

20.2 An LQ Optimal Observer Design Approach

Our study in the previous section clearly demonstrates the important role of an optimal observer in LQR or LQG optimal control. For our purpose of online optimisation of observers, we are going to deal with LQ optimal observer issues for LTI systems in this section. Our problem formulation is analogue to the LS fault estimation problem for LTV systems addressed in Chap. 8. On the other hand, for our purpose we will solve the optimal estimation problem using an alternative method which allows us to perform a cost function based online optimisation of an LTI observer.

20.2.1 Problem Formulation and the Basic Idea of the Solution

To simplify our study, we consider LTI systems described by

$$x\left(k+1\right) = Ax\left(k\right) + Ed_1(k) \in \mathcal{R}^n, \tag{20.20}$$

$$y(k) = Cx(k) + F_d d_2(k) \in \mathcal{R}^m. \tag{20.21}$$

Here, $d_1 \in \mathcal{R}^{k_{d_1}}, d_2 \in \mathcal{R}^{k_{d_2}}$ are l_2-norm bounded unknown input vectors, and E, F_d are known matrices of appropriate dimensions. Our estimation problem is formulated as: given measurement data, $y\left(k_0\right), \cdots, y(k)$, solve the optimisation problem described by

$$\min_{x(k_0),d_1,d_2} \frac{1}{2}\left(\|x(k_0)\|^2_{P_0^{-1}} + \|d_1\|^2_{2,[k_0,k]} + \|d_2\|^2_{2,[k_0,k]}\right), P_0 > 0, \tag{20.22}$$

$$\text{s.t. } x\left(i+1\right) = Ax\left(i\right) + Ed_1(i), \tag{20.23}$$

$$y(i) = Cx(i) + F_d d_2(i), i = k_0, \cdots, k. \tag{20.24}$$

By introducing Lagrange multipliers $\lambda_x\left(i\right) \in \mathcal{R}^n, \lambda_y\left(i\right) \in \mathcal{R}^m$, we re-write the above optimisation problem as

$$\min_{d_1,d_2,x(i)} J, \tag{20.25}$$

$$J = \left(\begin{array}{c} \frac{1}{2}\|x(k_0)\|^2_{P_0^{-1}} + \frac{1}{2}\sum_{i=k_0}^{k}\|d_1(i)\|^2 + \frac{1}{2}\sum_{i=k_0}^{k}\|d_2(i)\|^2 \\ + \sum_{i=k_0}^{k}\lambda_y^T\left(i\right)\left(y\left(i\right) - Cx\left(i\right) - F_d d_2(i)\right) \\ + \sum_{i=k_0}^{k}\lambda_x^T\left(i+1\right)\left(x\left(i+1\right) - Ax\left(i\right) - Ed_1(i)\right) \end{array}\right)$$

$$= \left(\begin{array}{c} \frac{1}{2}\|x(k_0)\|^2_{P_0^{-1}} + \frac{1}{2}\sum_{i=k_0}^{k}\|d_1(i)\|^2 + \frac{1}{2}\sum_{i=k_0}^{k}\|d_2(i)\|^2 \\ + \sum_{i=k_0}^{k}\lambda_y^T\left(i\right)\left(y\left(i\right) - Cx\left(i\right) - F_d d_2(i)\right) \\ - \sum_{i=k_0}^{k}\lambda_x^T\left(i+1\right)\left(Ax\left(i\right) + Ed_1(i)\right) + \sum_{i=k_0}^{k}\lambda_x^T\left(i\right)x\left(i\right) \\ -\lambda_x^T\left(k_0\right)x\left(k_0\right) + \lambda_x^T\left(k+1\right)x(k+1) \end{array}\right).$$

Solving

$$\frac{\partial J}{\partial x(k_0)}=0,\ \frac{\partial J}{\partial x(k+1)}=0,$$
$$\frac{\partial J}{\partial x(i)}=0,\ \frac{\partial J}{\partial d_1(i)}=0,\ \frac{\partial J}{\partial d_2(i)}=0,\ \frac{\partial J}{\partial \lambda_x(i+1)}=0,\ \frac{\partial J}{\partial \lambda_y(i)}=0,$$

for $i=k_0+1,\cdots,k$, yields

$$\begin{aligned}P_0^{-1}x\,(k_0)-\lambda_x(k_0)=0,\ \lambda_x(k+1)=0,\\ \lambda_x(i)-C^T\lambda_y(i)-A^T\lambda_x(i+1)=0,\\ \hat{d}_1(i)=E^T\lambda_x(i+1),\ \hat{d}_2(i)=F_d^T\lambda_y(i),\\ \hat{x}\,(i+1)=A\hat{x}\,(i)+E\hat{d}_1(i),\ y(i)=C\hat{x}\,(i)+F_d\hat{d}_2(i).\end{aligned}$$

It holds

$$\lambda_x(i)=A^T\lambda_x(i+1)+C^T\lambda_y(i),\ \lambda_x(k+1)=0, \tag{20.26}$$
$$F_dF_d^T\lambda_y(i)=y(i)-C\hat{x}\,(i)\,, \tag{20.27}$$
$$\hat{x}\,(i+1)=A\hat{x}\,(i)+EE^T\lambda_x(i+1),\ \hat{x}\,(k_0)=P_0\lambda_x(k_0). \tag{20.28}$$

We would like to remark that $\hat{x}(i)$, $\hat{d}_1(i)$ and $\hat{d}_2(i)$ are the estimates for $x(i)$, $d_1(i)$ and $d_2(i)$ given data $y\,(k_0)\,,\ y(k_0+1),\cdots,\ y(k)$. For the sake of simplicity, we adopt these notations for $\hat{x}(i\,|k\,)$, $\hat{d}_1(i\,|k\,)$ and $\hat{d}_2(i\,|k\,)$.

20.2.2 A Solution

We now solve (20.26)–(20.28), which build a $2n$-dimensional LTI system with couplings between the state and co-state variables $\hat{x}(i)$ and $\lambda_x(i)$, $\lambda_y(i)$, and is driven by $y(i)$. This brings us to assume that

$$\hat{x}(i)=P(i)\lambda_x(i)+\alpha(i) \tag{20.29}$$

with $P(i)$, $\alpha(i)$ to be determined. Consider

$$\hat{x}(i+1)=P(i+1)\lambda_x(i+1)+\alpha(i+1)$$

and (20.28). It yields

$$A\hat{x}\,(i)+EE^T\lambda_x(i+1)=P(i+1)\lambda_x(i+1)+\alpha(i+1). \tag{20.30}$$

From (20.26), (20.27) and (20.29) we have, on the other hand,

$$\begin{aligned}
&CP(i)A^T\lambda(i+1) + CP(i)C^T\lambda_y(i) + C\alpha(i) = y(i) - F_dF_d^T\lambda_y(i)\\
&\Longrightarrow \lambda_y(i) = R^{-1}(i)\left(y(i) - C\alpha(i) - CP(i)A^T\lambda_x(i+1)\right), \qquad (20.31)\\
&\hat{x}(i) = P(i)\left(A^T - C^TR^{-1}(i)CP(i)A^T\right)\lambda_x(i+1) \qquad (20.32)\\
&+ P(i)C^TR^{-1}(i)\left(y(i) - C\alpha(i)\right) + \alpha(i),\\
&R(i) = F_d^TF_d + CP(i)C^T.
\end{aligned}$$

Substituting $\hat{x}\,(i)$ into (20.30) gives

$$\begin{aligned}
&\left(AP(i)A^T + EE^T - AP(i)C^TR^{-1}(i)CP(i)A^T\right)\lambda_x(i+1)\\
&+ AP(i)C^TR^{-1}(i)\left(y(i) - C\alpha(i)\right) + A\alpha(i)\\
&= P(i+1)\lambda_x(i+1) + \alpha(i+1).
\end{aligned}$$

This results in

$$P(i+1) = AP(i)A^T + EE^T - L\,(i)\,R(i)L^T(i), \qquad (20.33)$$

$$\alpha(i+1) = A\alpha(i) + L\,(i)\left(y(i) - C\alpha(i)\right), \qquad (20.34)$$

$$L\,(i) = AP(i)C^TR^{-1}(i), \qquad (20.35)$$

and furthermore the boundary values

$$\hat{x}\,(k_0) = P_0\lambda_x(k_0) \Longrightarrow \alpha(k_0) = 0,\; P(k_0) = P_0.$$

It follows from the boundary condition in (20.26) that

$$\hat{x}(k+1) = \alpha(k+1).$$

Recall that $\hat{x}(k+1)$ is the one-step ahead prediction of $x(k+1)$ (using data up to k). Hence, (20.34) gives the one-step prediction (estimation) formula for $x(k)$,

$$\begin{aligned}
&\hat{x}(k+1\,|k) = A\hat{x}\,(k\,|k-1) + L\,(k)\left(y(k) - C\hat{x}\,(k\,|k-1)\right), \qquad (20.36)\\
&\hat{x}\,(k\,|k-1) = \alpha(k),
\end{aligned}$$

and (20.33) as well as (20.35) are the recursion and update forms for the computation of $P(k)$, $L\,(k)$, respectively. Because k could be any integer larger than k_0, this estimator is indeed identical with the LS observer studied in Chap. 8.

Note that $\hat{x}(i)$ given in (20.32) can be further written as

$$\begin{aligned}
\hat{x}(i) = {} & \alpha(i) + P(i)C^TR^{-1}(i)\left(y(i) - C\alpha(i)\right)\\
& + P(i)\left(A^T - C^TR^{-1}(i)CP(i)A^T\right)\lambda_x(i+1). \qquad (20.37)
\end{aligned}$$

Moreover, as shown in the example given below, the first two terms in the above equation are indeed the estimation for $x(i)$ using the data up to i. That is

$$\hat{x}(i\,|i\,) = \alpha(i) + P(i)C^T R^{-1}(i)\,(y(i) - C\alpha(i))\,.$$

Thus,

$$\hat{x}(i) = \hat{x}(i\,|i\,) + P(i)\left(A^T - C^T R^{-1}(i)CP(i)A^T\right)\lambda_x(i+1).$$

In summary, the estimations of $\hat{x}(i)$, $\hat{d}_1(i)$ as well as $\hat{d}_2(i)$ can be performed by the following algorithm:

Algorithm 20.1 *LQ observer*

Step 0: *Computation of $P(i), \alpha(i)$ (as one-step ahead prediction of $x(i)$) according to (20.33)-(20.35);*

Step 1: *Computation of $\hat{x}(i)$ according to (20.37) for given $\lambda_x(i+1), \alpha(i)$;*

Step 2: *Computation of $\lambda_y(i)$ according to (20.31) and*

$$\hat{d}_1(i) = E^T\lambda_x(i+1), \hat{d}_2(i) = F_d^T\lambda_y(i);$$

Step 3: *Computation of $\lambda_x(i)$ using (20.26), $i+1 \to i$ and go to Step 1.*

Example 20.1 *As an example, the computation of*

$$\hat{x}(k) = \hat{x}\,(k\,|k\,)$$

is illustrated. It follows from (20.32) and the relations

$$\lambda_x(k+1) = 0, \alpha(k) = \hat{x}\,(k\,|k-1\,)$$

that

$$\hat{x}(k) = \hat{x}\,(k\,|k-1\,) + P(k)C^T R^{-1}(k)\left(y(k) - C\hat{x}\,(k\,|k-1\,)\right),$$

which is identical with the LS estimate $\hat{x}\,(k\,|k\,)$ for $x(k)$. By this example, it can also be seen that

$$\hat{x}(k+1) = \hat{x}(k+1\,|k\,) = A\hat{x}\,(k\,|k\,)\,,$$

since

$$\begin{aligned}\hat{x}(k+1) &= P(k+1)\lambda_x(k+1) + \alpha(k+1), \lambda_x(k+1) = 0,\\ \hat{x}(k+1) &= A\hat{x}(k) + EE^T\lambda_x(k+1).\end{aligned}$$

Note that for $k_0 = -\infty$, the LTV (one-step prediction) observer becomes an LTI system given by

$$\begin{aligned}\hat{x}(k+1\,|k) &= A\hat{x}\,(k\,|k-1) + L\left(y(k) - C\hat{x}\,(k\,|k-1)\right),\\ L &= APC^T R^{-1},\ R = F_d F_d^T + CPC^T,\\ P &= APA^T + EE^T - LRL^T.\end{aligned}$$

20.2.3 The Dual Form

In the previous study, we have introduced the co-state vector $\lambda_x(k)$ as an auxiliary variable to solve the optimisation problem. The dynamics of $\lambda_x(k)$ and $\hat{x}(k)$ are coupled and build a $2n$ -dimensional system. In this sub-section, we derive an alternative form of approaching the optimisation problem, which is expressed in terms of the dynamics of $\lambda_x(k)$ and can be interpreted as the dual form of the LQ control problem.

We first consider (20.26) and re-write it, using relation (20.31), into

$$\lambda_x(i) = A^T\lambda_x(i+1) - C^T L^T\,(i)\,\lambda_x(i+1) + C^T R^{-1}(i)\,(y(i) - C\alpha(i)). \quad (20.38)$$

Moreover, by (20.27) and (20.31), we have

$$y(i) - C\hat{x}(i) = F_d F_d^T\left(R^{-1}(i)\,(y(i) - C\alpha(i)) - L^T\,(i)\,\lambda_x(i+1)\right). \quad (20.39)$$

Equations (20.38) and (20.39) reveal that

- the dynamic system (20.38) with $\lambda_x(i)$ as the state vector can be interpreted as closed-loop configured with $-L^T\,(i)\,\lambda_x(i+1)$ as feedback,
- analogue to the observer-based system model (20.5), system (20.38) is driven by the residual signal $y(i) - C\alpha(i)$ as well, and
- the feedback of $y(i) - C\hat{x}(i)$ in the observer consists of the feedback of $-L^T\,(i)$ $\lambda_x(i+1)$ and the residual signal $y(i) - C\alpha(i)$.

Remember further

$$\hat{d}_1(i) = E^T\lambda_x(i+1), \hat{d}_2(i) = F_d^T\lambda_y(i). \quad (20.40)$$

Hence, the co-state variables $\lambda_x(i), \lambda_y(i)$ can be viewed as a carrier of information about unknown input and uncertainties to be estimated. This is the further motivation of our subsequent work.

We are now in the position to formulate our LQ estimation problem as the dual form of LQ control. For our purpose, we consider steady state estimation towards an LTI optimal observer and thus set $k_0 = -\infty$. The cost function (20.22) becomes

$$\frac{1}{2}\left(\left\|E^T\lambda_x(i+1)\right\|_{2,(-\infty,k]}^2 + \left\|L^T\lambda_x(i+1)\right\|_{2,(-\infty,k]}^2\right)$$

and the LQ optimal estimation problem is formulated as

$$\min_{L} \frac{1}{2}\left(\left\|E^T\lambda_x(i+1)\right\|^2_{2,(-\infty,k-1]} + \left\|L^T\lambda_x(i+1)\right\|^2_{2,(-\infty,k-1]}\right) \tag{20.41}$$

$$\begin{aligned}\text{s.t. } \lambda_x(i) = A^T\lambda_x(i+1) - C^T L^T \lambda_x(i+1) + C^T R^{-1} r(i), \\ i \in (-\infty, k], \lambda(k+1) = 0.\end{aligned} \tag{20.42}$$

In (20.42), $r(i)$ represents $(y(i) - C\alpha(i))$, which, recalling $\alpha(i)$ being the one-step ahead prediction of $x(i)$, is the residual vector. When $d_1(i), d_2(i)$ are white noises, $r(i)$ is also white. It follows from our discussion in the last section that $r(i)$ has indeed no influence on the optimisation solution, and thus, for our discussion, $r(i)$ is assumed to be zero.

The optimisation problem (20.41)–(20.42) is the dual form to the optimal state feedback problem

$$\begin{aligned}&\min_{F} \frac{1}{2}\left(\|x(i)\|^2_{Q,2,[k,\infty)} + \|Fx(i)\|^2_{W,2,[k,\infty)}\right) \\ &\text{s.t. } x(i+1) = Ax(i) + Bu(i), u(i) = Fx(i), i \in [k, \infty),\end{aligned}$$

with the substitution in sense of the duality

$$A \to A^T, B \to C^T, F \to -L^T, \tag{20.43}$$

where Q, W are the weighting matrix equal to

$$Q = EE^T, W = F_d F_d^T.$$

The solution of this problem is well-known and given by

$$\begin{aligned}F &= -\left(W + B^T PB\right)^{-1} B^T PA, \\ P &= A^T PA + W - F^T\left(W + B^T PB\right) F.\end{aligned}$$

Thus, by means of the duality relations (20.43), we have exactly the optimal solution L given by

$$\begin{aligned}L &= APC^T\left(W + CPC^T\right)^{-1}, \\ P &= APA^T + Q - LRL^T.\end{aligned}$$

Below, as an example, we derive the solution for the optimisation problem (20.41)–(20.42) directly.

Example 20.2 *Let*

$$\begin{aligned}J(k) &= \left\| E^T \lambda(i+1) \right\|^2_{2,(-\infty,k-1]} + \left\| L^T \lambda(i+1) \right\|^2_{W,2,(-\infty,k-1]} \\ &= \sum_{i=-\infty}^{k-1} \lambda^T(i+1) \left(EE^T + LWL^T \right) \lambda(i+1) \\ &= J(k-1) + \lambda^T(k) \left(EE^T + LWL^T \right) \lambda(k), \\ W &= F_d F_d^T.\end{aligned}$$

Assume

$$J(k) = \lambda^T(k) P \lambda(k).$$

It holds

$$\begin{aligned}\lambda^T(k) P \lambda(k) &= \lambda^T(k-1) P \lambda(k-1) + \lambda^T(k) \left(EE^T + LWL^T \right) \lambda(k) \\ &= \lambda^T(k) \left(\left(A^T - C^T L^T \right)^T P \left(A^T - C^T L^T \right) + EE^T + LWL^T \right) \lambda(k).\end{aligned}$$

Now, minimising $\lambda^T(k) P \lambda(k)$ with respect to L leads to

$$L = APC^T \left(W + CPC^T \right)^{-1}.$$

Moreover,

$$\begin{aligned}P &= \left(A^T - C^T L^T \right)^T P \left(A^T - C^T L^T \right) + EE^T + LWL^T \\ &= APA^T + EE^T - L \left(W + CPC^T \right) L^T.\end{aligned}$$

This result is identical with the solution achieved by the duality given above.

It is worth remarking that the optimisation problem (20.41)–(20.42) can also be solved using the dual form of dynamic programming technique. It has been proved in Sect. 8.2 (referred to (8.51)) that

$$\begin{aligned}\min_L J(k) &= \min_L \sum_{i=-\infty}^{k-1} \lambda^T(i+1) \left(EE^T + LWL^T \right) \lambda(i+1) \\ &= \min_L \left(\lambda^T(k) \left(EE^T + LWL^T \right) \lambda(k) + \min_L J(k-1) \right).\end{aligned} \tag{20.44}$$

Equation (20.44) is the dual form of the well-known dynamic programming principle. In general, it can be written as

$$\min_{L(i),i\in[k_0,k-1]} J(k) = \min_{L(k-1)} \begin{pmatrix} \lambda^T(k)\left(EE^T + L(k-1)WL^T(k-1)\right)\lambda(k) \\ +\min_{L(i),i\in[k_0,k-2]} J(k-1) \end{pmatrix},$$

which results in an LTV observer.

20.2.4 LQ Observers for Systems with Input Vector

The simplified system model considered in the previous Sect. (20.20)–(20.21), is now extended to

$$x(k+1) = Ax(k) + Bu(k) + Ed_1(k), \tag{20.45}$$

$$y(k) = Cx(k) + Du(k) + F_d d_2(k), u(k) \in \mathcal{R}^p, \tag{20.46}$$

in order to include the influence of the system input vector $u(k)$. With the same design objective, we further adopt the cost function (20.22) with the constraints given by (20.45)–(20.46). Repeating the solution procedure presented in the last sub-sections results in the optimal observer. We summarise the main results as follows without providing detailed computations:

- one-step ahead optimal observer

$$\hat{x}(k+1\,|k) = A\hat{x}(k\,|k-1) + Bu(k) + L(k)r(k), \tag{20.47}$$

$$r(k) = y(k) - C\hat{x}(k\,|k-1) - Du(k), \tag{20.48}$$

where $L(k)$ is given below;
- recursive algorithm for $\hat{x}(i) = \hat{x}(i\,|k), \hat{d}_1(i) = \hat{d}_1(i\,|k), \hat{d}_2(i) = \hat{d}_2(i\,|k), i = k_0, \cdots, k,$

$$\begin{aligned}
\alpha(i+1) &= A\alpha(i) + Bu(i) + L(i)r(i), \\
\alpha(i) &= \hat{x}(i\,|i-1), r(i) = y(i) - C\alpha(i) - Du(i), \\
\hat{x}(i\,|i) &= \alpha(i) + P(i)C^T R^{-1}(i)r(i), \\
\hat{x}(i) &= \hat{x}(i\,|i) + P(i)\left(A^T - C^T R^{-1}(i)CP(i)A^T\right)\lambda_x(i+1), \\
\lambda_x(i) &= (A - CL(i))^T \lambda_x(i+1) + C^T R^{-1}(i)\left(y(i) - C\alpha(i)\right), \\
\lambda_y(i) &= R^{-1}(i)\left(y(i) - C\alpha(i) - CP(i)A^T\lambda_x(i+1)\right), \\
\hat{d}_1(i) &= E^T\lambda_x(i+1), \hat{d}_2(i) = F_d^T\lambda_y(i), \\
P(i+1) &= AP(i)A^T + EE^T - L(i)R(i)L^T(i), \\
L(i) &= AP(i)C^T R^{-1}(i), R(i) = F_d F_d^T + CP(i)C^T
\end{aligned}$$

with the boundary conditions

$$\alpha(k_0) = 0, \lambda_x(k+1) = 0, P(k_0) = P_0;$$

- some useful relations

$$\begin{aligned} \hat{x}(i) &= P(i)\lambda_x(i) + \alpha(i), \\ \hat{x}(i+1) &= A\hat{x}(i) + Bu(i) + EE^T\lambda_x(i+1), \hat{x}(k_0) = P_0\lambda_x(k_0). \end{aligned}$$

It is evident from the above equations that the co-state vector $\lambda_x(i)$ is a function of estimation errors caused by uncertainties in the system under supervision, for instance, the unknown input vectors d_1, d_2, and independent of $u(i)$.

20.3 LQ Control Performance Monitoring and Recovering

Having intensively studied the LQ control techniques for nominal systems with noises or disturbances, we begin in this section with our initial task: performance recovery and fault-tolerant control. The objective of this section is to propose a basic scheme for the LQ control performance recovery by updating the state feedback control gain F. We would like to emphasise that this scheme will be generally embedded in a fault-tolerant control system as a functionality module, although it can work independently.

20.3.1 Problem Formulation

We consider a (nominal) feedback control loop with the plant model (20.1) and a state feedback controller described by

$$\begin{aligned} \bar{x}(k+1) &= A\bar{x}(k) + B\bar{u}(k) + w(k), \bar{u}(k) = F_0\bar{x}(k) + v(k), \\ A &= A_0, B = B_0. \end{aligned}$$

Here, $w(k) \sim \mathcal{N}(0, \Sigma_w)$ is white process noise and uncorrelated with $\bar{x}(k)$, $v(k)$. It is assumed that the control system is operating in the steady state

$$\begin{aligned} \mathcal{E}\bar{x}(k+1) &= \mathcal{E}\left(A_{F_0}\bar{x}(k) + Bv(k) + w(k)\right) = \mathcal{E}\bar{x}(k) \Longrightarrow \\ \mathcal{E}\bar{x}(k) &= \left(I - A_{F_0}\right)^{-1} Bv(k) =: x_0, \\ v(k) &= v_o, A_{F_0} = A + BF_0. \end{aligned}$$

The dynamics of the closed-loop is governed by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k), x(k) = \bar{x}(k) - x_0, \\ u(k) &= F_0 x(k). \end{aligned} \tag{20.49}$$

As control performance, the quadratic cost function

$$\begin{aligned} J(i) &= \mathcal{E} \sum_{k=i}^{\infty} \gamma^{k-i} \left(x^T(k) Qx(k) + u^T(k) Ru(k) \right), \\ R &> 0, Q \geq 0, 0 < \gamma < 1, \end{aligned} \tag{20.50}$$

is under consideration. It is noteworthy that the cost function $J(i)$ is a prediction of the control performance for a given controller, also called control policy. It indicates which value of the control performance is to be expected when the actual controller (control policy) is continuously applied in the time interval $[i, \infty]$. Here, $k = \infty$ can be interpreted, in the engineering sense, as the end of a production process or a mission.

It is assumed that some faults or mismatching in the system cause changes in the system matrices A and B modelled by

$$A = A_0 + \Delta A, B = B_0 + \Delta B,$$

where $\Delta A, \Delta B$ are some unknown constant matrices. On the assumption that the closed-loop is asymptotically stable, the dynamics of the closed-loop becomes

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) + d_0, u(k) = F_0 x(k), \\ d_0 &= (\Delta A + \Delta B F_0) x_0 + \Delta B v_0. \end{aligned} \tag{20.51}$$

Next, we will study how to detect such changes and to update the controller (control policy) to be tolerant to them.

20.3.2 Reference-Model Based Detection of Performance Degradation

Following our discussions in the previous sections and remembering that $x(k)$ (in fact $\bar{x}(k)$) is a measurement variable, straightforward computations lead to the following value of the cost function during the fault-free (steady state) operation

$$\begin{aligned} J_{ref}(i) &= x^T(i) Px(i) + c, \\ P &= \gamma A_{F_0}^T P A_{F_0} + Q + F_0^T R F_0, c = \frac{\gamma tr\left(\Sigma_w P\right)}{1-\gamma}, A_{F_0} = A_0 + B_0 F_0. \end{aligned} \tag{20.52}$$

This value of the cost function is defined as the reference for monitoring performance degradation in the system under consideration. Performance degradation is detected, when

$$J(i) > J_{ref}(i).$$

Given a tolerance threshold J_{th}, an action of recovering performance degradation is to be activated, if

$$J(i) - J_{ref}(i) > J_{th} \ (> 0).$$

Remark 20.2 *In practical applications, it is realistic to set a constant reference like*

$$J_{ref} = \max_{i \in [k_1, k_2]} x^T(i) Px(i) + c, \tag{20.53}$$

where $[k_1, k_2]$ *is the time interval of interest and could be defined by the user.*

Next, an approach is proposed to evaluate performance degradation. To this end, assume that the cost function (20.50) is

$$\begin{aligned} J(i) &= \mathcal{E} \sum_{k=i}^{\infty} \gamma^{k-i} \left(x^T(k) Qx(k) + u^T(k) Ru(k) \right) \\ &= x^T(i) Px(i) + x^T(i) c_1 + c_2 \end{aligned} \tag{20.54}$$

with some constant vector $c_1 \in \mathcal{R}^n$ and constant c_2, and write it as

$$J(i) = \mathcal{E} \left(x^T(i) \left(Q + F_0^T R F_0 \right) x(i) + \gamma \begin{pmatrix} x^T(i+1) Px(i+1) \\ +x^T(i+1) c_1 + c_2 \end{pmatrix} \right). \tag{20.55}$$

It turns out, by taking into account (20.51) and some straightforward computations,

$$P = \gamma (A + BF_0)^T P (A + BF_0) + Q + F_0^T R F_0, \tag{20.56}$$

$$c_1 = 2\gamma \left(I - \gamma (A + BF_0)^T \right)^{-1} (A + BF_0)^T P d_0, \tag{20.57}$$

$$c_2 = \gamma \frac{tr(\Sigma_w P) + d_0^T P d_0 + d_0^T c_1}{1 - \gamma}. \tag{20.58}$$

Thus, (20.54) is a performance (degradation) prediction model with P, c_1, c_2 as the model parameters that are functions of unknown matrices $\Delta A, \Delta B$. In other words, in order to predict the performance degradation, P, c_1, c_2 should be online identified using measurement data. To this end, re-write (20.54) into

$$J(i) = x^T(i) Px(i) + x^T(i) c_1 + c_2 = \omega^T \phi(i), \tag{20.59}$$

where ω is the parameter vector including all parameters to be identified, $\phi(i)$ is a vector of time functions consisting of the process data. Concretely, ω is composed of $n(n+1)/2$ parameters of the $n \times n$ dimensional matrix P (as an SPD matrix), n parameters of c_1 and c_2 and hence

$$\omega = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_\eta \end{bmatrix} \in \mathcal{R}^\eta, \eta = (n+1)n/2 + n + 1 = (n+1)(n+2)/2.$$

The vector $\phi(i)$ is

$$\begin{aligned} \phi(i) &= \begin{bmatrix} \phi_1(i) \\ \vdots \\ \phi_\eta(i) \end{bmatrix} \in \mathcal{R}^\eta, \\ \phi_j(i) &\in \left\{1, x_q(i), q = 1, \cdots, n, x_q(i)x_r(i), q, r = 1, \cdots, n\right\}, j = 1, \cdots, \eta. \end{aligned}$$

Now, we are able to write (20.55) into the following form, based on which the parameter vector ω is identified,

$$\begin{aligned} \omega^T \phi(i) = x^T(i)\left(Q + F_0^T R F_0\right) x(i) + \gamma \omega^T \phi(i+1) \Longrightarrow \\ \omega^T\left(\phi(i) - \gamma\phi(i+1)\right) = x^T(i)\left(Q + F_0^T R F_0\right) x(i). \end{aligned} \tag{20.60}$$

Consequently, we are in a position to run the following algorithm for performance degradation monitoring and detection.

Algorithm 20.2 *Performance degradation monitoring and detection*

Step 0: *Compute $J_{ref}(i)$ or J_{ref} according to (20.52) or (20.53);*
Step 1: *Collect measurement data $x(i), x(i+1), \cdots, x(i+N+1)$;*
Step 2: *Form*

$$\begin{aligned} \Phi &= \left[\phi(i) - \gamma\phi(i+1) \cdots \phi(i+N) - \gamma\phi(i+N+1)\right], \\ \varphi &= \left[x^T(i)\left(Q + F_0^T R F_0\right) x(i) \cdots x^T(i+N)\left(Q + F_0^T R F_0\right) x(i+N)\right]; \end{aligned}$$

Step 3: *Run LS parameter estimation, for instance,*

$$\hat{\omega}^T = \varphi \Phi^T \left(\Phi\Phi^T\right)^{-1};$$

Step 4: *Compute*

$$J(i) = \hat{\omega}^T \phi(i);$$

Step 5: *If*

$$J(i) > J_{ref}(i) \Longrightarrow alarm,$$

otherwise go to Step 1.

Remark 20.3 *In order to achieve a good prediction, the number N is to be sufficiently large. In addition, sufficient excitation should be guaranteed for the LS estimation. Alternatively, regularised LS can be adopted. The recursive LS is also a practical solution.*

20.3.3 Performance Residual Based Detection of Performance Degradation

The online identification of the system performance model, as performed in Algorithm 20.2, delivers sufficiently accurate prediction of the system performance. On the other hand, the necessary online computation and time for collecting sufficient number of data could be, from the application viewpoint, problematic. Alternatively, we propose an approach based on the so-called the performance residual.

Suppose that the system under consideration is running under the normal operation condition with the nominal controller and, by collecting sufficient data, the cost function (performance) model (20.54) is identified. Note that this identification will be done one time and there is no real-time requirement on its performing. Recall that $J(i)$ can be written into a recursive form

$$J(i) = \mathcal{E}\left(x^T(i)\, Qx(i) + u^T(i)Ru(i) + \gamma J(i+1)\right).$$

This allows us to model the performance function by the following difference equation,

$$\begin{aligned} x^T(i)\, Px(i) - \gamma x^T(i+1)Px(i+1) + (x(i) - \gamma x(i+1))^T c_1 \\ + (1-\gamma)\, c_2 - x^T(i)\, Qx(i) - u^T(i)Ru(i) = 0, \end{aligned} \tag{20.61}$$

which is also called Bellman equation. On the basis of the above performance model, we introduce the definition of performance residual.

Definition 20.1 *Given the closed-loop system model (20.51) and the corresponding performance model (20.61), the signal* $r_P(i)$,

$$\begin{aligned} r_P(i) = x^T(i)\, Px(i) - \gamma x^T(i+1)Px(i+1) + (x(i) - \gamma x(i+1))^T c_1 \\ + (1-\gamma)\, c_2 - x^T(i)\, Qx(i) - u^T(i)Ru(i), \end{aligned} \tag{20.62}$$

is called performance residual, and the system (20.62) is called performance residual generator.

The performance residual generator is generally a nonlinear dynamic system that delivers the performance residual signal with slight variations around zero during normal operation. In order to detect performance degradation, it is expected that significant changes in $r_P(i)$ would be observed, when variations in the system matrices A and B are caused by performance degradation. According to (20.56)–(20.58), they will lead to variations in P, c_1, c_2 in the performance model (20.54). To be specific, let $\delta_{A_F}, \delta_{d_0}$ be the (unknown) changes in $A + BF_0$, d_0, and denote the corresponding solutions of P, c_1, c_2 by

$$\begin{aligned}
P + \Delta P =& \gamma \left(A + BF_0 + \delta_{A_F}\right)^T (P + \Delta P)\left(A + BF_0 + \delta_{A_F}\right) + Q + F_0^T RF_0, \\
& c_1 + \Delta c_1 = \\
& 2\gamma \left(I - \gamma \left(A + BF_0 + \delta_{A_F}\right)^T\right)^{-1} \left(A + BF_0 + \delta_{A_F}\right)^T (P + \Delta P)\left(d_0 + \delta_{d_0}\right), \\
& c_2 + \Delta c_2 = \frac{\gamma tr\left(\Sigma_w (P + \Delta P)\right)}{1-\gamma} \\
& + \gamma \frac{\left(d_0 + \delta_{d_0}\right)^T (P + \Delta P)\left(d_0 + \delta_{d_0}\right) + \left(d_0 + \delta_{d_0}\right)^T (c_1 + \Delta c_1)}{1-\gamma}.
\end{aligned}$$

It turns out

$$\begin{aligned}
\Delta P =& \gamma \left(A + BF_0 + \delta_{A_F}\right)^T \Delta P \left(A + BF_0 + \delta_{A_F}\right) + \gamma \delta_{A_F}^T P \delta_{A_F} \\
& + \gamma \delta_{A_F}^T P (A + BF_0) + \gamma (A + BF_0)^T P \delta_{A_F}, \\
& \frac{\Delta c_1}{2\gamma} = \left(I - \gamma \left(A + BF_0 + \delta_{A_F}\right)^T\right)^{-1} \begin{pmatrix} \delta_{A_F}^T (P + \Delta P)\left(d_0 + \delta_{d_0}\right) + \\ (A + BF_0)^T \Delta P \left(d_0 + \delta_{d_0}\right) + \\ (A + BF_0)^T P \delta_{d_0} \end{pmatrix} \\
& + \left(I - \gamma \left(A + BF_0 + \delta_{A_F}\right)^T\right)^{-1} \delta_{A_F}^T \left(I - \gamma (A + BF_0)^T\right)^{-1} (A + BF_0)^T P d_0, \\
& \Delta c_2 = \gamma \frac{tr\left(\Sigma_w \Delta P\right) + \left(d_0 + \delta_{d_0}\right)^T \Delta P \left(d_0 + \delta_{d_0}\right) + \delta_{d_0}^T P \delta_{d_0} + 2\delta_{d_0}^T P d_0}{1-\gamma} \\
& + \gamma \frac{d_0^T \Delta c_1 + \delta_{d_0}^T c_1 + \delta_{d_0}^T \Delta c_1}{1-\gamma}.
\end{aligned}$$

Finally, $r_P(i)$ satisfies

$$\begin{aligned}
r_P(i) = & x^T(i)\,\Delta P x(i) - \gamma x^T(i+1)\Delta P x(i+1) + (x(i) - \gamma x(i+1))^T \Delta c_1 \\
& + (1-\gamma)\,\Delta c_2.
\end{aligned} \tag{20.63}$$

For our purpose of building an evaluation function and determining the threshold correspondingly, we analyse the influence of ΔP, Δc_1, Δc_2 on $J(i)$. Write $r_P(i)$ as

$$\begin{aligned}r_P(i) &= \left(x^T(i) \otimes x^T(i) - \gamma x^T(i+1) \otimes x^T(i+1)\right) vec\left(\varDelta P\right)\\&\quad + \left(x(i) - \gamma x(i+1)\right)^T \varDelta c_1 + (1-\gamma)\varDelta c_2\\&\quad =: \varphi\left(x(i), x(i+1)\right)\phi\left(\varDelta\right)\end{aligned}$$

with

$$\begin{aligned}&\varphi\left(x(i), x(i+1)\right) =\\&\left[\, x^T(i) \otimes x^T(i) - \gamma x^T(i+1) \otimes x^T(i+1) \quad \left(x(i) - \gamma x(i+1)\right)^T \quad 1-\gamma \,\right],\\&\phi\left(\varDelta\right) = \begin{bmatrix} vec\left(\varDelta P\right) \\ \varDelta c_1 \\ \varDelta c_2 \end{bmatrix}.\end{aligned}$$

It yields

$$\begin{aligned}r_P^2(i) &= \varphi\left(x(i), x(i+1)\right)\phi\left(\varDelta\right)\phi^T\left(\varDelta\right)\varphi^T\left(x(i), x(i+1)\right)\\&\le \phi^T\left(\varDelta\right)\phi\left(\varDelta\right)\varphi\left(x(i),\; x(i+1)\right)\varphi^T\left(x(i), x(i+1)\right).\end{aligned} \tag{20.64}$$

This suggests to define the evaluation function as

$$J(i) = \frac{r_P^2(i)}{\varphi\left(x(i), x(i+1)\right)\varphi^T\left(x(i), x(i+1)\right)}. \tag{20.65}$$

Define $\Omega_{\varDelta P}$, $\Omega_{\varDelta c_1}$, $\Omega_{\varDelta c_2}$ as the value ranges of $\varDelta P$, $\varDelta c_1$, $\varDelta c_2$, which are accepted as (normal) operational variations and denote

$$\Omega_{\varDelta} := \left\{\Omega_{\varDelta P}, \Omega_{\varDelta c_1}, \Omega_{\varDelta c_2}\right\}.$$

Correspondingly, the threshold is set to be

$$J_{th} = \max_{\{\varDelta P, \varDelta c_1, \varDelta c_2\} \in \Omega_{\varDelta}} \phi^T\left(\varDelta\right)\phi\left(\varDelta\right), \tag{20.66}$$

since, according to (20.64) and (20.65),

$$J(i) \le \phi^T\left(\varDelta\right)\phi\left(\varDelta\right).$$

Consequently, the detection logic

$$\begin{cases} J(i) \le J_{th}, & \text{normal operation,} \\ J(i) > J_{th}, & \text{performance degradation,} \end{cases}$$

is adopted. Note that it is hard to solve (20.66) analytically. Alternatively, the RA-technique aided threshold setting algorithms introduced in Chap. 18, for instance, Algorithm 18.2, can be used.

20.3.4 Performance Recovery by Updating the State Feedback Gain

Once not allowed performance degradation is predicted, action to recover the performance will be activated. In this sub-section, we will study updating the feedback gain matrix F for this purpose. To be specific, we will find a solution for the following optimisation problem: given the feedback control loop

$$\begin{aligned} x\left(k+1\right) &= \left(A+BF_0\right)x\left(k\right) + B\Delta u(k) + w(k) + d_0, \\ \Delta u(k) &= Fx(k), \end{aligned}$$

find a feedback gain updating F so that cost function

$$J(i) = \mathcal{E}\sum_{k=i}^{\infty}\gamma^{k-i}\left(x^T\left(k\right)Q\left(k\right) + \left(Fx(k)\right)^T RFx(k)\right),$$

is minimised. The overall control loop will be optimised by adding an additional control signal

$$\Delta u(k) = \Delta Fx(k).$$

In other words, the total control input is

$$u(k) = F_0 x(k) + \Delta u(k) = \left(F_0 + \Delta F\right)x(k).$$

The idea for our solution is inspired by the so-called Q-learning known in the reinforcement learning based LQ-controller optimisation. Recall that the LQ optimal control gain should be

$$F = F_0 + \Delta F = -\gamma\left(R + \gamma B^T PB\right)^{-1} B^T PA, \tag{20.67}$$

$$P = \gamma A^T PA + Q - \gamma^2 A^T PB\left(R + \gamma B^T PB\right)^{-1} B^T PA \Longleftrightarrow \tag{20.68}$$

$$P = \gamma A_F^T PA_F + Q + F^T\left(R + \gamma B^T PB\right)F,\ P > 0, \tag{20.69}$$

in which B, A are, however, unknown. The key step to solve this problem is an iterative computation of the solution of Riccati equation (20.68) without knowledge of A and B. To this end, we propose the following scheme along the lines described by Lewis et al. in 2012 in their survey paper on the application of reinforcement learning technique to optimal adaptive control (the reference is given at the end of this chapter). We first introduce a known result that builds the theoretical basis of our solution.

Theorem 20.1 *Let $P_i, i = 0, 1, \cdots,$ be the solutions of*

$$P_i = A_{F_i}^T P_i A_{F_i} + Q + F_i^T RF_i,\ A_{F_i} = A + BF_i,\ i = 0, 1, \cdots, \tag{20.70}$$

where

$$F_{i+1} = -\left(R + B^T P_i B\right)^{-1} B^T P_i A, i = 0, 1, \cdots, \tag{20.71}$$

A_{F_0} should be a Schur matrix, A, B are the system matrices given in the system model (20.1) and R, Q are the weighting matrices adopted in the cost function (20.50). Then,

$$P \leq P_{i+1} \leq P_i \cdots, i = 0, 1, \cdots,$$
$$\lim_{i \to \infty} P_i = P,$$

where P is the solution of Riccati equation

$$P = A^T P A + Q - A^T P B \left(R + B^T P B\right)^{-1} B^T P A > 0. \tag{20.72}$$

This theorem was published by Hewer in 1971, and the corresponding reference is given at the end of this chapter. In order to implement the iterative algorithm (20.70)–(20.71) without knowledge of the system dynamics, we propose to add a test signal in the control loop to identify the needed parameters for building the control law (20.71) that converges to (20.67).

Remark 20.4 *Note that the result in the above theorem holds for any $\gamma \in (0, 1)$. In fact, $\forall \gamma \in (0, 1)$, the initial optimisation problem can be equivalently formulated as*

$$\min_F J(i) = \mathcal{E} \sum_{k=i}^{\infty} \left(x^T(k) Q(k) + (Fx(k))^T RFx(k)\right),$$
$$s.t.\ x(k+1) = \bar{A}x(k) + \bar{B}u(k) + w(k), \bar{A} = \sqrt{\gamma}A, \bar{B} = \sqrt{\gamma}B.$$

Let

$$\Delta u(k) = \vartheta(k) \Longrightarrow u(k) = F_j x(k) + \vartheta(k), j = 0, 1, \cdots, \tag{20.73}$$
$$\vartheta(k+1) = A_\vartheta \vartheta(k) + \varpi(k), \vartheta(0) = \vartheta_0, \tag{20.74}$$

where ϑ_0 is some constant (vector) as design parameter and $\varpi(k)$ is a white noise with

$$\mathcal{E}\varpi(k) = 0, \mathcal{E}\varpi(k)\varpi^T(k) = \Sigma_\varpi,$$

and independent of $x(k)$, $w(k)$, and A_ϑ is schur and can be set as a design parameter. Now, we write the overall system dynamics in the following compact form

$$\begin{bmatrix} x(k+1) \\ \vartheta(k+1) \end{bmatrix} = \begin{bmatrix} A + BF_j & B \\ 0 & A_\vartheta \end{bmatrix} \begin{bmatrix} x(k) \\ \vartheta(k) \end{bmatrix} + \begin{bmatrix} d_0 \\ 0 \end{bmatrix} + \begin{bmatrix} w(k) \\ \varpi(k) \end{bmatrix},$$

and consider the cost function

$$J_j(i) = \mathcal{E} \sum_{k=i}^{\infty} \gamma^{k-i} \left(x^T(k) Q_j x(k) + \vartheta^T(k) R \vartheta(k) \right), Q_j = Q + F_j^T R F_j.$$

Similar to (20.54)–(20.58), we have

$$\begin{aligned}
J_j(i) &= x^T(i) P_j x(i) + x^T(i) c_{1,j} + c_{2,j}, \; j = 0, 1, \cdots, \\
&= \left[x^T(i) \; \vartheta^T(i) \right] \left(\begin{bmatrix} P_j & P_{x\vartheta,j} \\ P_{x\vartheta,j}^T & P_{\vartheta,j} \end{bmatrix} \begin{bmatrix} x(i) \\ \vartheta(i) \end{bmatrix} + c_{1,j} \right) + c_{2,j}, \\
P_j &= \gamma A_{F_j}^T P_j A_{F_j} + Q_j, A_{F_j} = A + B F_j, && (20.75) \\
P_{x\vartheta,j} &= \gamma A_{F_j}^T \left(P_j B + P_{x\vartheta,j} A_\vartheta \right), && (20.76) \\
P_{\vartheta,j} &= \gamma \left(B^T P_j B + A_\vartheta^T P_{x\vartheta,j}^T B + B^T P_{x\vartheta,j} A_\vartheta + A_\vartheta^T P_{\vartheta,j} A_\vartheta \right) + R, && (20.77) \\
c_{1,j} &= 2\gamma \left(I - \gamma \begin{bmatrix} A_{F_j} & B \\ 0 & A_\vartheta \end{bmatrix}^T \right)^{-1} \begin{bmatrix} A_{F_j} & B \\ 0 & A_\vartheta \end{bmatrix}^T \begin{bmatrix} P_j \\ P_{x\vartheta,j}^T \end{bmatrix} d_0, && (20.78) \\
c_{2,j} &= \frac{\gamma}{1-\gamma} \left(tr \left(\begin{bmatrix} \Sigma_w & 0 \\ 0 & \Sigma_\varpi \end{bmatrix} \begin{bmatrix} P_j & P_{x\vartheta,j} \\ P_{x\vartheta,j}^T & P_{\vartheta,j} \end{bmatrix} \right) + d_0^T P_j d_0 + d_0^T \bar{c}_{1,j} \right), && (20.79) \\
\bar{c}_{1,j} &= 2\gamma \left(I - \gamma A_{F_j}^T \right)^{-1} A_{F_j}^T P_j d_0. && (20.80)
\end{aligned}$$

Moreover, the results on the identification of parameters P, c_1, c_2 in the cost function (20.54) and the associated Algorithm 20.1 presented in the last sub-section can be directly applied for identifying

$$\bar{P}_j = \begin{bmatrix} P_j & P_{x\vartheta,j} \\ P_{x\vartheta,j}^T & P_{\vartheta,j} \end{bmatrix}$$

and $c_{1,j}, c_{2,j}$ given in (20.78) and (20.79). Remember that, for building the feedback control gain matrix F_{j+1},

$$\begin{aligned}
F_{j+1} &= -\gamma \left(R + \gamma B^T P_j B \right)^{-1} B^T P_j A \\
&= -\gamma \left(R + \gamma B^T P_j B \right)^{-1} \left(B^T P_j A_{F_j} - B^T P_j B F_j \right),
\end{aligned}$$

matrices $B^T P_j B$, $B^T P_j A$ are needed, which are embedded in (20.76) and (20.80). We propose the following approximation as a solution.

Recall that A_ϑ is a design parameter (matrix). We set

$$A_\vartheta = \rho I, \tag{20.81}$$

and furthermore $|\rho|$ is sufficiently small,

$$|\rho| \ll 1, \tag{20.82}$$

so that

$$\begin{gathered}
B^T P_j B + A_\vartheta^T P_{x\vartheta,j}^T B + B^T P_{x\vartheta,j} A_\vartheta + A_\vartheta^T P_{\vartheta,j} A_\vartheta \approx B^T P_j B, \\
\gamma A_{F_j}^T \left(P_j B + P_{x\vartheta,j} A_\vartheta\right) \approx \gamma A_{F_j}^T P_j B.
\end{gathered}$$

As a result, the feedback control gain matrix is approximated by

$$\begin{aligned}
F_{j+1} &= -\gamma \left(R + \gamma B^T P_j B\right)^{-1} \left(B^T P_j A_{F_j} - B^T P_j B F_j\right) \\
&\approx -P_{\vartheta,j}^{-1} \left(P_{x\vartheta,j}^T - \left(P_{\vartheta,j} - R\right) F_j\right).
\end{aligned} \tag{20.83}$$

In fact, considering

$$\forall \mu > 0,\ B^T P_{x\vartheta,j} + P_{x\vartheta,j}^T B \leq \mu I + \frac{1}{\mu} B^T P_{x\vartheta,j} P_{x\vartheta,j}^T B,$$

it holds

$$\begin{aligned}
&\forall \varepsilon > 0, \exists \rho, \text{ so that } A_\vartheta^T P_{x\vartheta,j}^T B + B^T P_{x\vartheta,j} A_\vartheta = \rho \left(B^T P_{x\vartheta,j} + P_{x\vartheta,j}^T B\right) \\
&\leq \rho \left(\mu I + \frac{1}{\mu} B^T P_{x\vartheta,j} P_{x\vartheta,j}^T B\right) \leq \varepsilon I.
\end{aligned}$$

Moreover,

$$\gamma \left(A + B F_j\right)^T P_{x\vartheta,j} A_\vartheta = \gamma \rho \left(A + B F_j\right)^T P_{x\vartheta,j}.$$

Thus,

$$\lim_{\rho \to 0} \left(A_\vartheta^T P_{x\vartheta,j}^T B + B^T P_{x\vartheta,j} A_\vartheta\right) = 0,\ \lim_{\rho \to 0} \gamma \rho \left(A + B F_j\right)^T P_{x\vartheta,j} = 0.$$

It can be claimed that

$$\begin{aligned}
\lim_{\rho \to 0} F_{j+1} &= -\lim_{\rho \to 0} P_{\vartheta,j}^{-1} \left(P_{x\vartheta,j}^T - \left(P_{\vartheta,j} - R\right) F_j\right) \\
&= -\gamma \left(R + \gamma B^T P_j B\right)^{-1} B^T P_j A.
\end{aligned} \tag{20.84}$$

We summarise the main results on updating the feedback gain matrix according to (20.67) in the following algorithm.

Algorithm 20.3 *Update of feedback gain aiming at recovering performance degradation*

Step 0: *Input data: R, Q, F_0 (the existing controller to be updated), set $j = 0$ and the tolerance value β;*

Step 1-1: *Set A_ϑ according to (20.81)–(20.82) with a sufficiently small ρ and generate $\vartheta(k), k = i, \cdots, i + N + 1$, according to (20.74);*

Step 1-2: *Apply the control law*

$$u(k) = F_j x(k) + \vartheta(k)$$

to the process and collect data $x(k), k = i, \cdots, i + N + 1$;

Step 1-3: *Identify $\bar{P}_j$ using Algorithm 20.2 with data $x(k), \vartheta(k), k = i, \cdots, i + N + 1$;*

Step 1-4: *Set $j = j + 1$* and *the feedback control gain F_j according to (20.83);*

Step 1-5 *If*

$$\left\| F_j - F_{j-1} \right\|_2 > \beta,$$

go to Step 1-2, otherwise

Step 2: *Output the feedback control gain*

$$F = F_j.$$

Remark 20.5 *Recall that $\vartheta(k)$ is an additional test signal for the identification purpose. In real applications, $\vartheta(k)$ should be selected carefully, when updating of the control gain is performed during the system operation, so that the system operation will not be remarkably affected.*

As mentioned, our work is inspired by the Q-learning method towards (real-time) optimal adaptive LQ controller, which is also known in the literature as Q-learning method of reinforcement learning technique. Our scheme and the updating algorithm are different from the Q-learning algorithms published in the literature. In fact, the core of our work is the identification of the system performance as well as some related system matrices based on Bellman equation. In this regard, Step 1-1 to Step 1-2 in Algorithm 20.3 can be viewed as (control) performance monitoring. We notice different handlings in the published Q-learning algorithms, and summarise some of them as follows:

- Without considering noises, the cost function

  $$J(i) = \sum_{k=i}^{\infty} \gamma^{k-i} \left(x^T(k) Q x(k) + u^T(k) R u(k) \right)$$

  can be expressed in terms of the so-called Q-function as

  $$J(i) = \begin{bmatrix} x^T(i) & u^T(i) \end{bmatrix} \begin{bmatrix} \gamma A^T P A + Q & \gamma B^T P A \\ \gamma A^T P B & R + \gamma B^T P B \end{bmatrix} \begin{bmatrix} x(i) \\ u(i) \end{bmatrix}.$$

  One possibility to identify the kernel matrix

$$S = \begin{bmatrix} \gamma A^T PA + Q & \gamma B^T PA \\ \gamma A^T PB & R + \gamma B^T PB \end{bmatrix}$$

is the use of Bellman equation. Note that this is only possible when

$$u(i) = Fx(i). \tag{20.85}$$

As a result, we have

$$\begin{gathered} J(i) = \left(x^T(i)\,Qx(i) + u^T(i)Ru(i)\right) + \gamma J(i+1) \Longleftrightarrow \\ \varphi^T(i)S\varphi(i) - \gamma\varphi^T(i+1)S\varphi(i+1) = x^T(i)\,Qx(i) + u^T(i)Ru(i), \\ \varphi^T(i) = \left[x^T(i)\; u^T(i)\right]. \end{gathered}$$

It seems that $\gamma A^T PB$, $R+\gamma B^T PB$ could be identified using $\varphi(i)$. Unfortunately, due to the relation (20.85),$J(i)$ becomes

$$J(i) = x^T(i)\begin{pmatrix} \gamma A^T PA + Q + \gamma F^T B^T PA + \gamma A^T PBF \\ +F^T\left(R + \gamma B^T PB\right)F \end{pmatrix} x(i)$$

and thus a direct identification of $\gamma A^T PB$, $R + \gamma B^T PB$ is impossible. In fact, the relation (20.85),

$$\begin{bmatrix} x(i) \\ u(i) \end{bmatrix} = \begin{bmatrix} I \\ F \end{bmatrix} x(i),$$

implies that the excitation for the identification of the kernel matrix S is not sufficient.

- In order to solve this problem, it has been suggested to add noise $\theta(i)$ to the control signal,

$$u(i) = Fx(i) + \theta(i), \tag{20.86}$$

and apply $x(i)$, $u(i)$ to identify $\gamma A^T PB$, $R + \gamma B^T PB$. Unfortunately, there is no well-established rule for the selection of $\theta(i)$. A direct use of $\theta(i)$ in the form of (20.86) does not lead to the solution. This can be seen from the following discussion. Assume that

$$\begin{gathered} J(i) = \left[x^T(i)\;\theta^T(i)\right] S \begin{bmatrix} x(i) \\ \theta(i) \end{bmatrix} \Longrightarrow \\ J(i+1) = \left[x^T(i+1)\;\theta^T(i+1)\right] S \begin{bmatrix} x(i+1) \\ \theta(i+1) \end{bmatrix}. \end{gathered} \tag{20.87}$$

But, $J(i)$ cannot be expressed in terms of Bellman equation like

$$J(i) = \left(x^T(i)\,Qx(i) + u^T(i)Ru(i)\right) + \gamma J(i+1),$$

since $\theta(i+1)$ is not a (linear) mapping of $\theta(i)$. In other words, the assumption (20.87) does not hold.

- In the literature, an alternative scheme has been proposed, in which the kernel matrix S and the feedback gain matrix F are identified and determined in an iterative process. To be specific, the iterative algorithm is described schematically by

$$\begin{aligned} & \begin{bmatrix} x^T(i) \; u^T(i) \end{bmatrix} S^{j+1} \begin{bmatrix} x(i) \\ u(i) \end{bmatrix} = x^T(i) Qx(i) + u^T(i) Ru(i) \qquad (20.88) \\ & + \begin{bmatrix} x^T(i+1) \; \left(F^j x(i+1)\right)^T \end{bmatrix} S^j \begin{bmatrix} x(i+1) \\ F^j x(i+1) \end{bmatrix}, \\ & F^{j+1} = -\left(S_{22}^{j+1}\right)^{-1} S_{21}^{j+1}, \; j = 0, 1, \cdots, \end{aligned}$$

where j is the iteration number,

$$S^{j+1} = \begin{bmatrix} S_{11}^{j+1} & S_{12}^{j+1} \\ S_{21}^{j+1} & S_{22}^{j+1} \end{bmatrix}, \; S_{12}^{j+1} = \left(S_{21}^{j+1}\right)^T,$$

and for the sake of simplicity, γ is set equal to one. The core of this algorithm is the identification of S^{j+1} based on (20.88) using the process data $(x(i), u(i))$. Note that for the identification reason, $u(i)$ could be a random signal, but should not be set equal to $F^j x(i)$. It should be also noticed that (20.88) is not a Bellman equation. In other words,

$$\begin{bmatrix} x^T(i) \; u^T(i) \end{bmatrix} S^{j+1} \begin{bmatrix} x(i) \\ u(i) \end{bmatrix}$$

is not the true performance value until S^j converges, and thus it cannot be used for the performance monitoring purpose.

The above discussion illustrates also why system (20.74) has been introduced in our solution. On the other hand, we would like to emphasise that applying our solution $\gamma A^T PB$, $R + \gamma B^T PB$ can be satisfactorily identified thanks to the relation (20.84).

20.4 Real-Time Monitoring and Optimisation of Observers

This section is dedicated to monitoring and updating observers and observer-based residual generators. To be specific, we will focus on real-time optimisation of an observer to match changes in the system dynamics. The basis for our efforts is the dual form of the observer design presented in Sect. 20.2.

20.4.1 Problem Formulation and Basic Idea

Consider LTI systems described by (20.45)–(20.46) with

$$A = A_o, B = B_o, C = C_o, D = D_o$$

as nominal system matrices. Here, d_1, d_2 will be specified in the sequel. We first formulate our problems of assessing and recovering estimation performance in a broader sense. Let

$$\hat{x}(k+1\,|k) = A\hat{x}\,(k\,|k-1) + Bu(k) + Lr(k), \tag{20.89}$$

$$r(k) = y(k) - C\hat{x}\,(k\,|k-1) - Du(k) \tag{20.90}$$

be a (stable) optimal state observer that delivers a one-step ahead prediction of the state vector $x(k)$ and residual vector $r(k)$. Roughly speaking, the state estimation error,

$$e(k) = x(k) - \hat{x}\,(k\,|k-1)\,,$$

is the key indicator for the estimation performance of an observer. Since $e(k)$ is not measurable, its assessment during system operations is challenging. Consequently, detection of estimation performance degradation caused by changes in the system dynamics and, associated with it, recovery of the estimation performance degradation are problems remaining to be solved.

By our study on LQ optimal observers in Sect. 20.2, we have introduced the co-state vector λ_x and the associated dual system,

$$\begin{aligned} \lambda_x(i) &= A^T\lambda_x(i+1) - C^T L^T \lambda_x(i+1) + C^T R^{-1} r(i), \\ \lambda_x(k+1) &= 0, \\ r(i) &= y(i) - C\hat{x}\,(i\,|i-1) - Du(i), i = k_0, \cdots, k, \end{aligned} \tag{20.91}$$

and demonstrated that

- driven by the residual vector r, system (20.91) is the information carrier about the uncertainties in the system expressed in terms of d_1, d_2,
- the LS estimations for d_1, d_2 are linear mappings of the co-state vector λ_x (as well as λ_y),
- the optimal observer problem can be equivalently expressed in terms of an LQ regulation problem with the dual system (20.91).

These results and conclusions inspire us to propose applying the dual system (20.91) for monitoring estimation performance and detecting performance degradation. In this context, we call system (20.91) estimation performance observer. We propose

$$J(k) = \sum_{i=-\infty}^{k} \lambda_x^T(i) Q \lambda_x(i),\ Q > 0 \tag{20.92}$$

as the general form of the cost function for performance assessment, and formulate the assessment and monitoring tasks as

- determination of the nominal value of the cost function (20.92) and
- development of performance monitoring algorithms with respect to the cost function (20.92).

A further task deals with updating the observer to recover the estimation performance, when strong performance degradation is detected. It is supposed that the estimation performance degradation is caused by $d_1(k)$, $d_2(k)$ which are either l_2-norm bounded unknown inputs or modelled by

$$\begin{aligned} d_1(k) &= \Delta Ax(k) + \Delta Bu(k) + w(k), \\ d_2(k) &= \Delta Cx(k) + \Delta Du(k) + q(k), \end{aligned}$$

where ΔA, ΔB, ΔC, ΔD are some unknown constant matrices representing changes in the system matrices A, B, C, D and $w(k)$, $q(k)$ are noises. The basic idea behind the performance degradation algorithm is the application of the LQ observer for estimating ΔA, ΔB, ΔC, ΔD.

20.4.2 Monitoring and Detection of Estimation Performance Degradation

Nominal Performance

As a reference for assessing estimation performance, we first define the (optimal) operation conditions and, under them, determine the reference value. For this purpose, let the reference system model be the following state space realisation,

$$\begin{aligned} x\,(k+1) =& A_o x\,(k) + B_o u(k) + \bar{w}(k), x(0) = x_0, & (20.93) \\ y(k) =& C_o x(k) + D_o u(k) + \bar{q}(k), & (20.94) \\ \bar{w}(k) =& E w(k) \sim \mathcal{N}\left(0, \bar{\Sigma}_w\right), \bar{\Sigma}_w = E \Sigma_w E^T, w(k) \sim \mathcal{N}\left(0, \Sigma_w\right), \\ \bar{q}(k) =& F_d q(k) \sim \mathcal{N}\left(0, \bar{\Sigma}_q\right), \bar{\Sigma}_q = F_d \Sigma_q F_d^T > 0, q(k) \sim \mathcal{N}\left(0, \Sigma_q\right), \end{aligned}$$

$$\mathcal{E}\left(\begin{bmatrix} w(i) \\ q(i) \\ x\,(0) \end{bmatrix}\begin{bmatrix} w(j) \\ q(j) \\ x\,(0) \end{bmatrix}^T\right) = \begin{bmatrix} \begin{bmatrix} \Sigma_w & 0 \\ 0 & \Sigma_q \end{bmatrix}\delta_{ij} & 0 \\ 0 & \Pi_0 \end{bmatrix}.$$

It is reasonable to run a Kalman filter,

$$\hat{x}(k+1\,|k) = A_o \hat{x}\,(k\,|k-1) + B_o u(k) + L_o r(k), \tag{20.95}$$

$$r(k) = y(k) - C_o \hat{x}\,(k\,|k-1) - D_o u(k), \tag{20.96}$$

$$P = A_o P A_o^T + \bar{\Sigma}_w - L_o \left(\bar{\Sigma}_q + C_o P C_o^T \right) L_o^T, \tag{20.97}$$

$$L_o = A_o P C_o^T \left(\bar{\Sigma}_q + C_o P C_o^T \right)^{-1}, \tag{20.98}$$

for an optimal estimation of the state vector and residual generation. It follows from our study in Sect. 20.2 that the estimation performance observer,

$$\begin{aligned} &\lambda(i) = A_o^T \lambda(i+1) - C_o^T L_o^T \lambda(i+1) + C_o^T R^{-1} r(i), \lambda(k+1) = 0, \\ &r(i) = y(i) - C_o \hat{x}\left(i \left| i-1 \right.\right) - D_o u(i), R = \bar{\Sigma}_q + C_o P C_o^T, i = k_0, \cdots, k, \end{aligned} \tag{20.99}$$

can be applied for the assessment of the estimation and monitoring performance.

Remark 20.6 *To simplify the notation, $\lambda(i)$ is adopted for $\lambda_x(i)$. Since we only consider the nominal system model without disturbance in the output model, it will not cause any confusion.*

Remember that the residual vector $r(i)$ is white noise satisfying

$$r(i) \sim \mathcal{N}\left(0, \left(\bar{\Sigma}_q + C_o P C_o^T\right)\right).$$

It is also clear that $\lambda(i+1), r(i)$ are uncorrelated. Now, we define the cost function as

$$J(k) = \mathcal{E} \sum_{i=-\infty}^{k} \gamma^{k-i} \lambda^T(i) \left(\bar{\Sigma}_w + L_o \bar{\Sigma}_q L_o^T \right) \lambda(i), 0 < \gamma < 1. \tag{20.100}$$

In order to determine the performance value $J(k)$, assume

$$J(k) = \lambda^T(k) P \lambda(k) + c, P > 0. \tag{20.101}$$

It holds

$$\begin{aligned} &J(k) = \mathcal{E}\left(\lambda^T(k) \left(\bar{\Sigma}_w + L_o \bar{\Sigma}_q L_o^T \right) \lambda(k) + \gamma J(k-1) \right) \Longrightarrow \\ &\lambda^T(k) P \lambda(k) + c = \\ &\mathcal{E}\left(\lambda^T(k) \left(\begin{array}{c} \gamma \left(A_o^T - C_o^T L_o^T \right)^T P \left(A_o^T - C_o^T L_o^T \right) \\ + \bar{\Sigma}_w + L_o \bar{\Sigma}_q L_o^T \end{array} \right) \lambda(k) \right) + \gamma tr\left(C_o P C_o^T R^{-1} \right). \end{aligned}$$

Thus, it is evident that P in (20.101) is the solution of the Riccati equation (20.97) and

$$c = \frac{\gamma tr\left(C_o P C_o^T R^{-1} \right)}{1 - \gamma}.$$

As a result, the performance value given in (20.101) is the nominal value and considered as a reference.

Performance degradation detection

When the system under consideration operates around the optimal operating point, the estimation performance can be calculated by means of (20.101). On the other hand, the cost function $J(k)$ is the solution of the following difference equation,

$$J(k) = \mathcal{E}\left(\lambda^T(k)\left(\bar{\Sigma}_w + L_o\bar{\Sigma}_q L_o^T\right)\lambda(k) + \gamma J(k-1)\right). \tag{20.102}$$

Substituting relation (20.101) into (20.102) yields

$$\lambda^T(k)\left(P - \bar{\Sigma}_w - L_o\bar{\Sigma}_q L_o^T\right)\lambda(k) - \gamma\lambda^T(k-1)P\lambda(k-1) + (1-\gamma)c = 0.$$

Now, we define r_P,

$$\begin{aligned} r_P(k) = \lambda^T(k)\left(P - \bar{\Sigma}_w - L_o\bar{\Sigma}_q L_o^T\right)\lambda(k) \\ -\gamma\lambda^T(k-1)P\lambda(k-1) + (1-\gamma)c \end{aligned} \tag{20.103}$$

as the (estimation) performance residual signal. It is clear that any change in the system will cause r_P differing from zero. In this way, degradation in the estimation performance caused by changes in the system is detected. For the real-time realisation, we propose the algorithm given below.

Algorithm 20.4 *Detection of estimation performance degradation*

Step 0: *Compute and save* P, c *based on the model (* 20.93*)–(*20.94*)*;
Step 1: *Run estimation performance observer (*20.99*) and collect* $\lambda(k-1), \lambda(k)$*;*
Step 2: *Compute performance residual* r_P *according to (*20.103*);*
Step 3: *Run the detection logic*

$$\begin{cases} J_{th,low} \le r_P(k) \le J_{th,high} \Longrightarrow fault - free \Longrightarrow go\ to\ Step 1, \\ \text{otherwise}, \textit{faulty}, \end{cases}$$

where $J_{th,low}, J_{th,high}$ *are thresholds, which should be set depending on system operation conditions.*

Alternatively, the control performance degradation detection method presented in Sect. 20.3.3 can be adopted for the same purpose.

20.4.3 Performance Degradation Recovery

Problem formulation

Consider the system model

$$x(k+1) = Ax(k) + Bu(k) + \bar{w}(k) = A_o x(k) + B_o u(k) + Ed_1(k), \tag{20.104}$$

$$y(k) = Cx(k) + Du(k) + \bar{q}(k) = C_o x(k) + D_o u(k) + F_d d_2(k) \tag{20.105}$$

with $d_1(k), d_2(k)$ being either l_2-norm bounded unknown inputs or modelled by

$$\begin{aligned} d_1(k) &= \varDelta_A x(k) + \varDelta_B u(k) + w(k), \\ d_2(k) &= \varDelta_C x(k) + \varDelta_D u(k) + q(k). \end{aligned}$$

Here, $w(k), q(k)$ are noises as defined previously, and

$$\varDelta = \begin{bmatrix} \varDelta_A & \varDelta_B \\ \varDelta_C & \varDelta_D \end{bmatrix}$$

represents uncertainties in the system matrices. Suppose that d_1, d_2 cause estimation performance degradation (in sense of our discussion in the previous sub-section) and triggers updating the Kalman filter (20.95) aiming at recovering performance degradation. In the sequel, we assume that sufficient process data, $y(i), u(i), i = k_0, \cdots, k$, have been collected, and E, F_d in the model (20.104)–(20.105) are known, and

$$rank \begin{bmatrix} E & 0 \\ 0 & F_d \end{bmatrix} = \dim \left(\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \right).$$

Recall the following equations achieved during our study on LQ optimal observers in Sect. 20.2:

$$\hat{x}\left(i+1\left|i\right.\right) = A_o \hat{x}\left(i\left|i-1\right.\right) + B_o u(i) + Lr(i), \tag{20.106}$$

$$r(i) = y(i) - C_o \hat{x}\left(i\left|i-1\right.\right) - D_o u(i), \tag{20.107}$$

$$\hat{x}(i\left|i\right.) = \hat{x}\left(i\left|i-1\right.\right) + PC_o^T Rr(i), \tag{20.108}$$

$$\hat{x}(i) = \hat{x}(i\left|i\right.) + P\left(A_o^T - C_o^T R^{-1} C_o P A_o^T\right) \lambda_x(i+1), \tag{20.109}$$

$$\lambda_x(i) = \left(A_o - LC_o\right)^T \lambda_x(i+1) + C_o^T R^{-1} r(i), \lambda_x(k+1) = 0, \tag{20.110}$$

$$\lambda_y(i) = R^{-1} r(i) - L^T \lambda_x(i+1), \tag{20.111}$$

$$\hat{d}_1(i) = E^T \lambda_x(i+1), \hat{d}_2(i) = F_d^T \lambda_y(i), \tag{20.112}$$

for $i = k_0, \cdots, k$, where

$$\begin{aligned} &L = A_o P C_o^T R^{-1}, P = A_o P A_o^T + EE^T - LRL^T, \\ &R = F_d F_d^T + C_o P C_o^T, \end{aligned}$$

and the standard notations

$$\hat{x}(i) = \hat{x}(i\left|k\right.),$$

$\hat{x}(i\,|i\,)$ as well as $\hat{x}(i\,|i-1\,)$ represent the estimates of $x(i)$ using the data $\{y(k_0),\cdots,y(k)\}$, and $\{y(k_0),\cdots,y(i)\}$ as well as $\{y(k_0),\cdots,y(i-1)\}$, respectively. Our tasks are

- to analyse the estimation performance of $\hat{x}(i)$, $\hat{d}_1(i)$ and $\hat{d}_2(i)$ delivered by the LQ observers,
- to apply them for the control performance degradation recovery, and
- in case of $d_1(i), d_2(i)$ representing model uncertainties, to estimate Δ, on account of the model

$$\begin{bmatrix} d_1(i) \\ d_2(i) \end{bmatrix} = \begin{bmatrix} \Delta_A & \Delta_B \\ \Delta_C & \Delta_D \end{bmatrix} \begin{bmatrix} x(i) \\ u(i) \end{bmatrix} + \begin{bmatrix} w(i) \\ q(i) \end{bmatrix}, \tag{20.113}$$

and using data

$$\hat{d}_1(i+j), \hat{d}_2(i+j), \hat{x}(i+j), u(i+j), j=0,1,\cdots,N, [i,i+N] \subset [k_0,k]\,.$$

Estimation performance analysis

For our estimation purpose, we first analyse the performance of $\hat{d}_1(i), \hat{d}_2(i)$ delivered by the LQ observer. It follows from (20.106)–(20.112) that

$$\lambda_x(i) = (A_o - LC_o)^T\, \lambda_x(i+1) + C_o^T R^{-1} r(i), \lambda_x(k+1) = 0, \tag{20.114}$$

$$\begin{bmatrix} \hat{d}_1(i) \\ \hat{d}_2(i) \end{bmatrix} = \begin{bmatrix} E^T \\ -F_d^T L^T \end{bmatrix} \lambda_x(i+1) + \begin{bmatrix} 0 \\ F_d^T R^{-1} \end{bmatrix} r(i). \tag{20.115}$$

That is, $\hat{d}_1(i), \hat{d}_2(i)$ are the output of a dynamic system with $\lambda_x(i+1)$ as its state vector and residual $r(i)$ as its input. It is straightforward to write $\hat{d}_1(i), \hat{d}_2(i)$ as

$$\begin{aligned}
\begin{bmatrix} \hat{d}_1(i) \\ \hat{d}_2(i) \end{bmatrix} =& E_L^T \left(A_L^T\right)^{s-1} \lambda_x(i+s) + \sum_{j=1}^{s-1} E_L^T \left(A_L^T\right)^{j-1} \bar{C}^T r(i+j) + \bar{F}_d^T R^{-1} r(i) \\
=& E_L^T \left(A_L^T\right)^{s-1} \lambda_x(i+s) + H_{r,s-1} \bar{r}_{s-1}(i+s-1), \\
A_L =& A_o - LC_o, E_L^T = \begin{bmatrix} E^T \\ -F_d^T L^T \end{bmatrix}, \bar{F}_d^T = \begin{bmatrix} 0 \\ F_d^T R^{-1/2} \end{bmatrix}, \bar{C}^T = C_o^T R^{-1/2}, \\
H_{r,s-1} =& \begin{bmatrix} \bar{F}_d^T & E_L^T \bar{C}^T & E_L^T A_L^T \bar{C}^T & \cdots & E_L^T \left(A_L^T\right)^{s-2} \bar{C}^T \end{bmatrix}, \\
\bar{r}_{s-1}(i+s-1) =& \begin{bmatrix} R^{-1/2} r(i) \\ \vdots \\ R^{-1/2} r(i+s-1) \end{bmatrix}.
\end{aligned}$$

On the other hand, it is known that

$$
\begin{aligned}
e\left(i+1\right) &= A_L e(i) + \begin{bmatrix} E & -LF_d \end{bmatrix} \begin{bmatrix} d_1(i) \\ d_2(i) \end{bmatrix}, \\
r(i) &= C_o e(i) + \begin{bmatrix} 0 & F_d \end{bmatrix} \begin{bmatrix} d_1(i) \\ d_2(i) \end{bmatrix}, e(i) = x(i) - \hat{x}\left(i\left|i-1\right.\right), \\
\bar{r}_{s-1}(i+s-1) &= \varGamma_{s-1} e(i) + H_{d,s-1} d_{s-1}(i+s-1), \\
\varGamma_{s-1} &= \begin{bmatrix} R^{-1/2} C_o \\ R^{-1/2} C_o A_L \\ \vdots \\ R^{-1/2} C_o A_L^{s-1} \end{bmatrix}, H_{d,s-1} = \begin{bmatrix} \bar{F}_d & 0 & \cdots & 0 \\ R^{-1/2} C_o E_L & \bar{F}_d & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ R^{-1/2} C_o A_L^{s-2} & \cdots & R^{-1/2} C_o E_L & \bar{F}_d \end{bmatrix},
\end{aligned}
$$

$$
d_{s-1}(i+s-1) = \begin{bmatrix} d(i) \\ \vdots \\ d(i+s-1) \end{bmatrix}, d(j) = \begin{bmatrix} d_1(j) \\ d_2(j) \end{bmatrix}.
$$

Suppose that i is the time instant before the uncertainty has caused significant changes in the system dynamics. Hence, it is reasonable to assume

$$
e(i) \approx 0. \tag{20.116}
$$

Moreover, for large s,

$$
\left(A_L^T\right)^{s-1} \approx 0.
$$

As a result, we have

$$
\begin{bmatrix} \hat{d}_1(i) \\ \hat{d}_2(i) \end{bmatrix} \approx H_{r,s-1} H_{d,s-1} d_{s-1}(i+s-1). \tag{20.117}
$$

Note the relations

$$
\begin{gathered}
\begin{bmatrix} E & 0 \end{bmatrix} \begin{bmatrix} 0 \\ F_d^T \end{bmatrix} = 0, \begin{bmatrix} 0 & F_d \end{bmatrix} \begin{bmatrix} 0 \\ F_d^T \end{bmatrix} = F_d F_d^T, \begin{bmatrix} E & 0 \end{bmatrix} \begin{bmatrix} E^T \\ 0 \end{bmatrix} = E E^T, \\
\begin{bmatrix} E & 0 \end{bmatrix} - L \begin{bmatrix} 0 & F_d \end{bmatrix} = \begin{bmatrix} E & -LF_d \end{bmatrix},
\end{gathered}
$$

which allow us to write L, R and the Riccati equation equivalently as

$$L = \left(A_o P C_o^T + \begin{bmatrix} E & 0 \end{bmatrix} \begin{bmatrix} 0 \\ F_d^T \end{bmatrix} \right) R^{-1}, \tag{20.118}$$

$$R = \begin{bmatrix} 0 & F_d \end{bmatrix} \begin{bmatrix} 0 \\ F_d^T \end{bmatrix} + C_o P C_o^T, \tag{20.119}$$

$$P = A_o P A_o^T + \begin{bmatrix} E & 0 \end{bmatrix} \begin{bmatrix} E^T \\ 0 \end{bmatrix} - L R L^T. \tag{20.120}$$

It follows from the result given in Lemma 7.1 that

$$H_{d,s-1} H_{d,s-1}^T = I, \tag{20.121}$$

when L, R are set according to (20.118) and P solves the Riccati equation (20.120). Thus, if $H_{d,s-1}$ is a square matrix, which is equivalent with

$$\dim(y) = \dim(d) = m,$$

it holds

$$H_{d,s-1}^T H_{d,s-1} = I. \tag{20.122}$$

Notice that

$$H_{r,s-1} = H_{d,s-1}^T (1 : m, :)$$

is the first m rows of $H_{d,s-1}^T$. It yields

$$H_{r,s-1} H_{d,s-1} = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \Longrightarrow \begin{bmatrix} \hat{d}_1(i) \\ \hat{d}_2(i) \end{bmatrix} = \begin{bmatrix} d_1(i) \\ d_2(i) \end{bmatrix}.$$

Remark 20.7 *In the study on Lemma 7.1, the initial condition is taken into account by extending the unknown input vector $d(k_1, k_2)$ to $\bar{d}(k_1, k_2)$ with the corresponding matrix $H_{\bar{d},o}(k_1, k_2)$. Its influence is then expressed by means of P_o, the initial condition of the Riccati equation (20.120). In our above result, this influence is directly included in the solution of the Riccati equation in form of P_o due to the formulation of the optimisation problem (20.25).*

It is straightforward that (20.122) still holds true by an output transformation, when

$$\dim(y) = m > \dim(d).$$

In case

$$\dim(y) < \dim(d),$$

matrix $H_{d,s-1}^T$ is in fact a pseudo-inverse of $H_{d,s-1}$ due to relation (20.121). In this sense, $H_{r,s-1} H_{d,s-1}$ is an LS approximation of matrix $\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}$. Considering

$$\begin{bmatrix} \hat{d}_1(i) \\ \hat{d}_2(i) \end{bmatrix} - \begin{bmatrix} d_1(i) \\ d_2(i) \end{bmatrix} \approx \left(H_{r,s-1} H_{d,s-1} - \begin{bmatrix} I & 0 \cdots 0 \end{bmatrix} \right) d_{s-1}(i+s-1),$$

it is suggested to check if

$$\left\| H_{r,s-1} H_{d,s-1} - \begin{bmatrix} I & 0 \cdots 0 \end{bmatrix} \right\|_F \leq \gamma$$

holds, before $\begin{bmatrix} \hat{d}_1(i) \\ \hat{d}_2(i) \end{bmatrix}$ is adopted for our estimation purpose. Here, $\gamma > 0$ is the tolerance defined by user.

Remark 20.8 *The above discussion also reveals another aspect (interpretation) of the dynamic system (20.114)–(20.115) with $\lambda_x(i+1)$ as its state vector, residual $r(i)$ as the input and $\hat{d}_1(i), \hat{d}_2(i)$ as the output. The dynamics of this system is in fact the first $k_d = \dim(d)$ rows of the pseudo-inverse of $H_{d,s-1}$. Recall that the proof of Lemma 7.1, (20.122) holds true only if the recursion*

$$P(i+1) = AP(i)A^T + EE^T - L(i)R(i)L^T(i)$$

converges to the constant matrix P. Moreover, the assumption (20.116) holds for $k_0 \to -\infty$. As a result, a good estimate for $d(i)$ is achievable only for $i \to -\infty$.

We now briefly address the estimates for the state vector $x(i)$. They are, as given in (20.106)–(20.112),

$$\begin{aligned} \hat{x}(i+1\,|i) &= A_o \hat{x}(i\,|i-1) + B_o u(i) + Lr(i), \\ \hat{x}(i\,|i) &= \hat{x}(i\,|i-1) + PC_o^T Rr(i), \\ \hat{x}(i) &= \hat{x}(i\,|i) + P\left(A_o^T - C_o^T R^{-1} C_o P A_o^T\right) \lambda_x(i+1). \end{aligned} \tag{20.123}$$

It is clear that $\hat{x}(i)$, different from $\hat{x}(i\,|i), \hat{x}(i+1\,|i)$, not only depends on the current and past process data $y(i-j), u(i-j), j = 0, 1, \cdots$, but also on the "future" process data $y(i+j), u(i+j), j = 1, \cdots, k-i$, as it is driven by $\lambda_x(i+1)$. In other words, $\hat{x}(i)$ delivers the best estimate for $x(i)$ among these three estimates. It is of interest to notice that $\hat{x}(i)$ is computed using Algorithm 20.1 from $i = k$ to $i = k_0$ downwards and, for instance, according to (20.123), instead of (20.28). On the other hand, multiplying A_o to both sides of (20.123) leads to

$$A_o \hat{x}(i) = A_o \hat{x}(i\,|i) + A_o P\left(A_o^T - C_o^T R^{-1} C_o P A_o^T\right) \lambda_x(i+1),$$

and further by Riccati equation (20.118)

$$A_o \hat{x}(i) = A_o \hat{x}(i\,|i) + \left(P - EE^T\right) \lambda_x(i+1).$$

Finally, according to

$$\hat{x}(i+1) = P\lambda_x(i+1) + \hat{x}\left(i+1\left|i\right.\right), \hat{x}\left(i+1\left|i\right.\right) = A_o\hat{x}(i\left|i\right.),$$

we have

$$\hat{x}(i+1) = A_o\hat{x}(i) + EE^T\lambda_x(i+1) = A_o\hat{x}(i) + E\hat{d}_1(i),$$

which is (20.28). From these computations it becomes clear that $\hat{x}(i)$ gives a good estimation of $x(i)$, when $d(i)$ is well estimated. Moreover, we would like to call the reader's attention to Remark 20.8, which tells us the best estimation $\hat{d}(i)$ and $\hat{x}(i)$ is achieved for $i \to -\infty$. In other words, by the backwardly recursive computation of $\hat{x}(i)$, the estimation performance will become better with the increasing number of the iterations.

An application of an LQ observer for recovering control performance degradation

It is known and also demonstrated in Sect. 20.1 that the $\mathcal{H}_2$ optimal control can be solved by optimising an LQ regulator and an LQ observer, separately and respectively. In Sect. 20.3, Algorithm 20.3 has been introduced for the online optimisation of the LQ regulator on the assumption of availability of the state vector $x(k)$ but without knowledge of the variations in the system model. Below, we give a modified version of Algorithm 20.3, in which $x(k)$ is replaced by $\hat{x}\left(i\left|k\right.\right)$ delivered by an LQ observer.

Algorithm 20.5 *Update of feedback gain based on the LQ state estimation*

Step 0: *Set A_ϑ according to (20.81)–(20.82) with a sufficiently small ρ and generate $\vartheta(k), k = i, \cdots, i+N+1$, according to (20.74);*

Step 1-1: *Apply the control law (20.73) to the process and collect $y(k), u(k), k = i, \cdots, i+N_1+1, \cdots, i+N_2$;*

Step 1-2: *Run LQ observer (20.106)–(20.112) using the data $y(k), u(k), k = i, \cdots, i+N_2$ and save*

$$\hat{x}(i) = \hat{x}(i\left|i+N_2\right.), \cdots, \hat{x}(i+1+N_1) = \hat{x}(i+1+N_1\left|i+N_2\right.);$$

Step 1-3: *Identify $\bar{P}$ using Algorithm 20.2 with data $\hat{x}(k), \vartheta(k), k = i, \cdots, i+N_1+1$;*

Step 2: *Repeat Step 1-1 to Step 1-3 until the update of the state feedback gain converges*;

Step 3: *Output the feedback control gain (20.84).*

It is followed by our discussion in the end of the last sub-section that N_2 should be sufficiently large so that $x(k), k = i, \cdots, i+N_1+1, N_1 << N_2$, can be well estimated. It is of considerable interest to notice that the optimisation of both LQ regulator and LQ observer is performed without perfect model knowledge.

Identification and Recovery Algorithm

Next, we introduce an algorithm for the estimation of uncertainty Δ on the basis of the model (20.113), once $\hat{d}_1(i), \hat{d}_2(i), \hat{x}(i)$ have been estimated with satisfactory performance. To this end, sufficient number of data are first collected to build

$$\Psi_d = \begin{bmatrix} \hat{d}_1(i) \cdots \hat{d}_1(i+N) \\ \hat{d}_2(i) \cdots \hat{d}_2(i+N) \end{bmatrix}, \Psi_{xu} = \begin{bmatrix} \hat{x}(i) \cdots \hat{x}(i+N) \\ u(i) \cdots u(i+N) \end{bmatrix}.$$

On the assumption that Ψ_{xu} has full row rank, which means sufficient excitation, an LMS estimate for Δ is given by

$$\hat{\Delta} = \begin{bmatrix} \hat{\Delta}_A & \hat{\Delta}_B \\ \hat{\Delta}_C & \hat{\Delta}_D \end{bmatrix} = \Psi_d \Sigma_{wq}^{-1} \Psi_{xu}^T \left(\Psi_{xu} \Sigma_{wq}^{-1} \Psi_{xu}^T \right)^{-1}, \qquad (20.124)$$

$$\Sigma_{wq} = \begin{bmatrix} \Sigma_w & 0 \\ 0 & \Sigma_q \end{bmatrix}.$$

Next, an update of Kalman filter (20.95)–(20.97) is performed, using the estimated $\hat{\Delta}$, as follows

$$\hat{x}(i+1|i) = \hat{A}\hat{x}(i|i-1) + \hat{B}u(i) + Lr(i), \qquad (20.125)$$

$$r(i) = y(i) - \hat{C}\hat{x}(i|i-1) - \hat{D}u(i), \qquad (20.126)$$

$$L = \hat{A}P\hat{C}^T \left(\bar{\Sigma}_q + \hat{C}P\hat{C}^T \right)^{-1}, P = \hat{A}P\hat{A}^T + \bar{\Sigma}_w - L\left(\bar{\Sigma}_q + \hat{C}P\hat{C}^T \right) L^T,$$

$$\hat{A} = A_o + E\hat{\Delta}_A, \hat{B} = B_o + E\hat{\Delta}_B, \hat{C} = C_o + F_d\hat{\Delta}_C, \hat{D} = D_o + F_d\hat{\Delta}_D.$$

It is followed by a check of the estimation performance, for instance, by means of Algorithm 20.4 with estimation performance observer

$$\lambda(i) = \hat{A}^T\lambda(i+1) - \hat{C}^T L^T \lambda(i+1) + \hat{C}^T R^{-1} r(i), \lambda(k+1) = 0,$$
$$R = \bar{\Sigma}_q + \hat{C}P\hat{C}^T, i = k_0, \cdots, k.$$

In case that the performance requirement is not satisfied, the above steps will be repeated. Below is the summary of the above performance degradation recovering procedure.

Algorithm 20.6 *Recovery of estimation performance degradation*

Step 0: *Set* $j = 0$ *and*

$$A_j = A_o, B_j = B_o, C_j = C_o, D_j = D_o;$$

Step 1: *Run LQ observer (20.106)–(20.112) by substituting* A_o, B_o, C_o and D_o *with* A_j, B_j, C_j *and* D_j, *and collect sufficient data*

$$\hat{d}_1(i+j), \hat{d}_2(i+j), \hat{x}(i+j), u(i+j), j = 0, 1, \cdots, N, [i, i+N] \subset [k_0, k];$$

Step 2: *Build* Ψ_d, Ψ_{xu} *and compute LMS estimate of* $\hat{\Delta}$ *according to (20.124);*

Step 3: *Set* $j = j + 1$,

$$
\begin{aligned}
&A_j = A_{j-1} + E\hat{\Delta}_A,\, B_j = B_{j-1} + E\hat{\Delta}_B,\, C_j = C_{j-1} + F_d\hat{\Delta}_C,\\
&D_j = D_{j-1} + F_d\hat{\Delta}_D,
\end{aligned}
$$

and run Kalman filter (20.125)–(20.126) by substituting $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ with A_j, B_j, C_j and D_j;

Step 4: *Run Algorithm 20.4 and check if the estimation performance requirement is satisfied. If yes$\Longrightarrow$ stop, otherwise go to Step 1.*

20.5 Notes and References

This chapter serves for four purposes,

- reviewing the standard LQG and LQR (or $\mathcal{H}_2$) control problems and providing the alternative solutions based on the observer-based input–output model (19.5)–(19.6), which allows us to handle control performance degradation monitoring and recovery problems separately as LQ controller and LQ observer optimisation,
- formulating and solving performance degradation monitoring and recovering problems for feedback control loops with an LQ controller,
- formulating LQ optimal observer design problem, studying its solution and some relevant issues, and finally
- formulating and solving LQ observer performance degradation monitoring and recovering problems.

Although our reviewing study on LQG and LQR/$\mathcal{H}_2$ control issues has been performed on the basis of the observer-based input–output model (19.5)–(19.6), the well-known methods and algorithms for dealing with LQ or LQG/LQR problems, for instance, the ones given in [1, 2], have been applied.

As a dual form of the LQ control scheme, the so-called LQ optimal observer has been introduced directly after the section on LQG and LQR control issues. It should be emphasised that the objective of this work is to establish a framework for monitoring and recovering of the observer-based estimation performance degradation, instead of addressing observer optimisation issues. In fact, from the viewpoint of the cost function, LQ optimal observer or estimation problem is similar to the LS observer defined by [3] and studied in Chap. 8. On the other hand, different from the study on LS observer in Chap. 8, in which the focus is on the solution of the one-step ahead prediction of the state and unknown input vectors, our major attention of the work on LQ optimal estimation is paid to

- the role and interpretation of the co-state vector $\lambda_x(k)$,
- the (smoothing) estimates of the state and unknown input vectors, $\hat{x}\left(i\,|k\right)$, $\hat{d}\left(i\,|k\right), i = k_0, \cdots, k$, and
- the dual form and relations between the LQ optimal controller and observer.

On account of this work, the LQ optimal estimation can be equivalently expressed in terms of an optimisation problem consisting of (i) a cost function with $\lambda_x(k)$ as its variables, and (ii) a dynamic system with $\lambda_x(k)$ as the state vector and the residual signal as the input. This re-formulated optimisation problem is the basis for the monitoring and recovering of observer performance degradation.

In the following two sections, performance degradation monitoring and recovering issues for control and estimation systems are investigated, respectively.

In order to detect control performance degradation, two algorithms have been proposed: (i) reference model-based approach, and (ii) performance residual model-based approach. In the first approach, an LTI system with a state feedback controller and driven by white process and measurement noises is first defined as the reference model. The quadratic performance value is computed according to

$$J_{ref}(i) = x^T(i)\, Px(i) + c$$

with a constant c and $P > 0$ as the solution of a Lyapunov equation. It should be kept in mind that the performance (degradation) value is in fact a prediction of the performance function

$$J(i) = \mathcal{E} \sum_{k=i}^{\infty} \gamma^{k-i} \left(x^T(k)\, Qx(k) + u^T(k) Ru(k) \right).$$

That means, what is to be assessed is the performance development beginning from the current time instant, when the current controller is continuously in use. For the online assessment of the performance degradation, the degradation model (20.54) is adopted and re-written into a regression model which enables an online identification of the parameter vector of the regression model and thus prediction of the performance degradation. This approach is computationally involved and requires collecting sufficient process data online.

The second approach is based on the degradation model (20.54) and the fact that the performance value $J(i)$ can also be written into a recursive form

$$J(i) = \mathcal{E} \left(x^T(i)\, Qx(i) + u^T(i) Ru(i) + \gamma J(i+1) \right).$$

They allow us to model performance degradation by the difference equation (20.61) and further to build the performance residual generation model (20.62). With the help of an analysis of the residual dynamics with respect to the variations in the performance model (20.61), residual evaluation function (20.65) is finally defined. It is evident that the needed online computation for the implementation of the second approach is, in comparison with the first approach, considerably less involved. Also, no significant detection delay is expected, since the detection algorithm is performed at each time instant without collecting a great number of data.

We would like to call the reader's attention to the essential role of the degradation model (20.54) and the corresponding difference equation (20.61) in our performance degradation detection schemes. In fact, the difference equation (20.61) is the Bellman equation [4] well-known in optimal control theory. This model provides us with a powerful computation tool to deal with issues like performance value prediction. Indeed, (system) performance value computation is a key step in the reinforcement learning technique that is well established in the machine learning theory. Since years, this technique has been drawing considerable research attention devoted to its application to real-time learning and optimisation in engineering systems [5]. In this context, the detection schemes proposed in this chapter can also be embedded in a learning-based online controller optimisation procedure. It is worth mentioning that in the reinforcement learning framework the performance residual $r_P(i)$ defined in (20.62) is called temporal difference (TD) and used for the value function updates [5, 6].

Our work dedicated to the control performance degradation recovery is inspired by the so-called Q-learning method, which is a well-established method in the reinforcement learning technique [5] and widely applied in online and model-free optimisation of LQ controllers [7]. Although we have applied the Q-learning method for recovering control performance degradation, the basic idea and the relevant mathematical handlings are similar. We have followed the ideas described by Lewis et al. in their survey paper [6] and proposed a scheme for updating state feedback gain matrix, once significant control performance degradation is detected. In this scheme, Theorem 20.1 published by Hewer in [8] plays a key role and builds the theoretical basis for updating the state feedback gain. Our work has focused on iteratively solving the associated Riccati equation using the process data. For our purpose, a method has been proposed to add certain noise for the identification of the kernel matrix needed for building the optimal control law. As discussed in the end of Sect. 20.3.4, this work is necessary. In fact, it is common knowledge that, for the identification of the kernel matrix, additional noise should be injected into the input [6]. Nevertheless, few results have been reported on the issues like

- how to create the noise and, above all,
- which signal should be used for the identification purpose.

In our work, we have proposed to generate the noise using the dynamic system (20.73)–(20.74) and proved that the influence of the noise on the control performance can be arbitrarily reduced to an acceptable level. Moreover, we have illustrated that the injected noise, instead of the input signal, should be used to identify the kernel matrix.

It should be noticed that the Hewer's iterative algorithm can also be, alternatively and without knowledge of the system dynamics, realised by a direct identification of the system matrices A and B, instead of the identification of the kernel matrix implemented in the Q-learning method. In fact, on the assumption of the system model

$$x(k+1) = Ax(k) + Bu(k) + w(k),$$

matrices A and B can be well identified using the available data $(x(k), u(k))$, $k = i, i+1, \cdots, i+N+1$, and LS estimation algorithm,

$$\begin{aligned}\begin{bmatrix}\hat{A} & \hat{B}\end{bmatrix} &= X(i+1)\, Z^T(i)\left(Z(i)Z^T(i)\right)^{-1},\\ X(i+1) &= \begin{bmatrix}x(i+1) & \cdots & x(i+N+1)\end{bmatrix},\\ Z(i) &= \begin{bmatrix}x(i) & \cdots & x(i+N)\\ u(i) & \cdots & u(i+N)\end{bmatrix}.\end{aligned}$$

From the viewpoint of the needed online computations, we know for $n > p+1$, which is generally the case in practice,

$$\frac{(n+p+1)(n+p)}{2} = \frac{n(n+p)}{2} + \frac{(p+1)(n+p)}{2} < n(n+p).$$

Here, $\frac{(n+p+1)(n+p)}{2}$, $n(n+p)$ are the numbers of the parameters to be identified for the kernel matrix and the matrices A and B, respectively. Nevertheless, the identification of the kernel matrix in the Q-learning should be performed repeatedly, while the identification of A and B can be theoretically realised by performing the LS algorithm one time. On the other hand, the solution P_j of the Lyapunov equation (20.70) is needed by each iteration when the matrices A and B are used, while in the Q-learning iteration, the control gain matrix is built directly using the sub-matrices of the identified kernel matrix.

From the viewpoint of recovering control performance degradation, the Q-learning algorithm proposed in our work is of two important advantages:

- a performance monitoring is performed at each iteration, and
- the performance recovery is realised step by step.

Indeed, these properties also allow us to stop the iteration running in Algorithm 20.3, as far as the required system performance is satisfied. For this reason, we prefer the use of the performance degradation recovering scheme proposed in this chapter.

The last part of our work addresses the monitoring and performance recovering issues of observers which are applied both for the control and fault detection purposes. To our best knowledge, there are rarely investigations dedicated to such topics. One key issue is how to define the (performance) cost function. It seems that the estimation errors of the state and unknown input vectors would be the reasonable and logic candidates. Unfortunately, they are not directly measurable. On account of our discussions in Sect. 20.2, we have proposed, alternatively, the cost function (20.92) with the co-state vector $\lambda_x(i)$ as variables. The vector $\lambda_x(i)$ consists of the state variables of the dynamic system (20.91) that is driven by the residual vector. Thanks to the duality, the detection issue of observer performance degradation can be then formulated and solved analogue to detecting LQ control performance degradation, as described in Algorithm 20.4.

The observer performance (degradation) recovery is, in its core, an estimation problem. The LQ observer delivers an estimation for variations in the system parameters. In most of applications, for instance, in real-time control, the user is interested in the estimates for $x(k)$, $d(k)$ using the process data up to the time instant k. In our study on the estimation performance, we have demonstrated that

- the (optimal) estimates for $x(i), d(i), i = \cdots, 0, 1, \cdots, k$, using data up to the time instant k, are delivered by the dynamic system (20.114)–(20.115),
- the estimation accuracy will increase with i converging to $-\infty$, and
- under certain conditions, the best estimates are achieved with $i = -\infty$, as pointed out in Remark 20.8.

On account on these facts, we have proposed Algorithms 20.5 and 20.6 for updating the feedback gain based on the LQ state estimation and recovering estimation performance degradation.

References

1. G. Chen, G. Chen, and S.-H. Hsu, *Linear Stochastic Control Systems*. CRC Press, 1995.
2. V. Kucera, *Analysis and Design of Discrete Linear Control Systems*. Prentice Hall, 1991.
3. J. Willems, "Deterministic least squares filtering," *Journal of Econometrics*, vol. 118, pp. 341–373, 2004.
4. R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.
5. R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
6. F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control," *IEEE Control Systems Magazine*, pp. 76–105, 2012.
7. F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, pp. 32–50, 2009.
8. G. A. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Trans. on Automatic Control*, vol. 16, pp. 382–384, 1971.

Chapter 21
Performance-Based Fault-Tolerant Control and Performance Recovery

21.1 Motivation and Basic Ideas for Performance-Based FTC

With steadily increasing demands for high product quality and production reliability as well as safety in industrial processes, FTC has received considerable attention in recent years, in both research and industrial application domains. This trend is well reflected by the great number of publications on FTC methods, some of them are given at the end of this chapter. Most of the FTC schemes are model-based and can be classified as

- passive FTC, in which the controller is designed as *a prior* to be robust against potential faults,
- active FTC, in which the controller parameters or algorithms are online adapted or even the controller is (online) re-configured, triggered by alarms or information of some detected faults.

The active FTC schemes promise

- optimal performance during the fault-free process operations, and
- efficient management of faulty process operations and recovering the system performance (up to certain required level).

On the other hand, due to the system complexity and often high real-time requirements, the design and implementation of an active FTC system are in general a challenging task. For these reasons, investigations on active FTC methods build the mainstream in the research field.

It can be observed that the major attention of the existing FTC systems including the embedded fault diagnosis unit are mainly dedicated to faults in process (hardware) components like sensors and actuators. By means of redundant components with the same or similar functionality or applying certain compensation technique, the functionality of the faulty component will be recovered or partially recovered

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_21

so that the overall system performance degradation is limited. We call these methods component oriented FTC. It should be mentioned that in automation industry, considerable efforts have been made in the recent decade to increase the component reliability and, more recently, to enhance the intelligent degree of those key system components. Smart sensors and actuators are nowadays the state of the art. And the new generation of smart system components are of the ability being self-diagnosis and self-repair.

In this and the next chapters, we study the so-called performance-based fault-tolerant control strategies. The major differences to the component oriented FTC schemes are summarised as follows:

- performance degradation detection is integrated into the FTC system and triggers FTC algorithms (action),
- in this context, any process operation and operation conditions, which cause unacceptable system performance degradation, are viewed as faults. Note that not only those component faults like sensor or actuator faults, but also, for instance, mismatching between system or controller parameters, changes in operation conditions, could cause performance degradation,
- the objective of the FTC scheme to be addressed is to recover the system performance to an acceptable level. For this reason, we often use the term *performance recovery* instead of FTC.

As a consequence of this FTC strategy, we will focus on those faults which could cause changes in the system dynamics like multiplicative or parametric faults. Also, we will not strictly distinguish between model uncertainties and faults. All those changes in the system under consideration will be viewed as uncertainties as far as they will not cause strong performance degradation. Once the system performance degradation approaches the unacceptable level, the system operation is viewed as faulty.

Although various types of system performances could be considered in the performance-based FTC framework, we only focus on those mostly considered control performances expressed in terms of system stability or indices introduced in Sect. 19.3.

The basic ideas of establishing a performance-based FTC framework are illustrated in Fig. 21.1 schematically. One of the key issues and cornerstones of the framework is monitoring and detection of performance degradation, which triggers the performance recovery and is also a part of the performance recovering process. To this end, a performance degradation model is to be established, whose input variables are process measurements and parameters depending on process operations and operation conditions. This allows us

- to assess the system performance real-time,
- to detect changes in system performance (performance degradation),

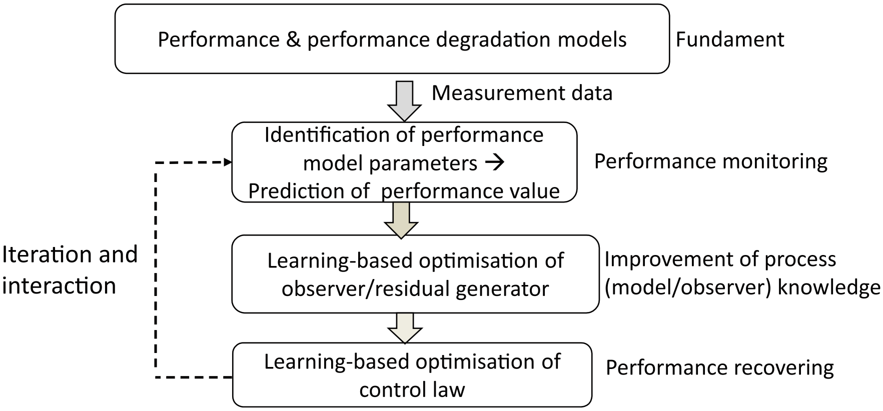

Performance & performance degradation models
Fundament
Measurement data
Identification of performance model parameters → Prediction of performance value
Performance monitoring
Iteration and interaction
Learning-based optimisation of observer/residual generator
Improvement of process (model/observer) knowledge
Learning-based optimisation of control law
Performance recovering

Fig. 21.1 Performance-based FTC: the schematic procedure

- to estimate the actual value of the performance (degradation) by identifying the model parameters after the changes in system performance have been detected, and
- based on which, the controller is optimised aiming at recovering the performance degradation.

It is worth remarking that the optimisation (update) of the controller will be realised in an iterative procedure, in which the performance (value) as the cost function is optimised with respect to the controller.

21.2 An Approach of Recovering Stability Margin Degradation

Stability is a fundamental characteristic of any feedback control system. Correspondingly, guaranteeing system stability is an essential task of all feedback controllers. In many industrial applications, in particular in those safety relevant sectors, the stable process operation mode is often referred as *fail-safe,* which is the ultimate requirement on a fault-tolerant control system.

Roughly speaking, stability margin is a quantitative indicator for the assessment of the stability of a feedback control system. As discussed in Sect. 19.3, changes in the plant of a feedback control system could cause degradation of the stability margin. Real-time recovering the stability margin degradation is of considerable practical interest. This is the motivation of our work in this section towards developing an approach of recovering stability margin degradation. Recall our conclusion in Sect. 19.3 that (i) recovering stability margin, (ii) loop performance degradations, and (iii) optimal fault detection can be achieved in a unified manner. It is expected that the approach to be developed in this section will simultaneously result in recovering the control loop performance degradation and enhancing the fault detectability. It is worth mentioning that our study in this section is strongly related to the work in Sect. 19.3. In fact, this section is a continuation of our study in Sect. 19.3, in which an approach will be proposed for real-time controller optimisation aiming at recovering stability and loop performance degradations as well as increasing fault detectability.

21.2.1 Preliminaries and Problem Formulation

Process description Consider the standard feedback control loop presented in Fig. 21.2 with $G(z)$ as the plant model, $K(z)$ as the feedback controller, $d \in \mathcal{R}^m$ denoting the possible stochastic noises or deterministic disturbances, and $v \in \mathcal{R}^p$ representing the reference signal or the output of a feed-forward controller driven by the reference signal.

Let the nominal model that describes fault-free system operations be

$$y(z) = G(z)u(z) + d(z), \, y \in \mathcal{R}^m, u \in \mathcal{R}^p \tag{21.1}$$

with minimal state space realisation

$$G = (A, B, C, D)\,, A \in \mathcal{R}^{n\times n}, B \in \mathcal{R}^{n\times p}, C \in \mathcal{R}^{m\times n}, D \in \mathcal{R}^{m\times p}.$$

The LCF and RCF of $G(z)$ are respectively given by

$$G(z) = \hat{M}_o^{-1}(z)\hat{N}_o(z) = N_o(z)M_o^{-1}(z), \tag{21.2}$$

and $\left(\hat{M}_o, \hat{N}_o\right)$ and (M_o, N_o) are left and right coprime pairs over $\mathcal{RH}_\infty$. Correspondingly, there exist $\hat{X}_o, \hat{Y}_o, X_o, Y_o \in \mathcal{RH}_\infty$ of appropriate dimensions and satisfying the Bezout identity. According to Youla parameterisation, any stabilising controller can then be parameterised by

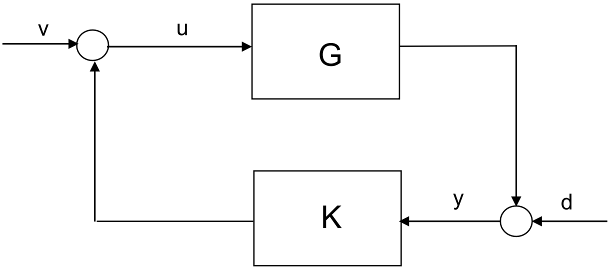


Fig. 21.2 Schematic description of the feedback control loop under consideration

$$K(z) = -U(z)V^{-1}(z) = -\hat{V}^{-1}(z)\hat{U}(z),$$
$$\begin{bmatrix} \hat{V} & \hat{U} \end{bmatrix} = \begin{bmatrix} X_o - Q\hat{N}_o & Y_o + Q\hat{M}_o \end{bmatrix}, \begin{bmatrix} V \\ U \end{bmatrix} = \begin{bmatrix} \hat{X}_o - N_o Q \\ \hat{Y}_o + M_o Q \end{bmatrix}$$

with the parameterisation matrix $Q(z) \in \mathcal{RH}_\infty$.

We model faulty process operations by

$$y(z) = G_f(z)u(z) + d(z) = \hat{M}^{-1}(z)\hat{N}(z)u(z) + d(z), \tag{21.3}$$

where $G_f(z)$ denotes the faulty plant model with $\hat{M} \in \mathcal{RH}_\infty^{m\times m}$, $\hat{N} \in \mathcal{RH}_\infty^{m\times p}$. It is evident that by defining

$$\begin{bmatrix} \Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} = \begin{bmatrix} \hat{N} - \hat{N}_o & \hat{M} - \hat{M}_o \end{bmatrix}, \tag{21.4}$$

the faulty plant (21.3) can be described by

$$G_f = \left(\hat{M}_o + \Delta_{\hat{M}_o}\right)^{-1}\left(\hat{N}_o + \Delta_{\hat{N}_o}\right). \tag{21.5}$$

It is obvious that

$$\begin{bmatrix} \Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \in \mathcal{RH}_\infty.$$

In Sect. 9.1, we have studied different forms of model uncertainties and illustrated that the coprime factor form (21.5) is representative and does not lead to loss of generality. Here, we would like to emphasise that in the context of monitoring and detecting performance degradation, $\Delta_{\hat{M}_o}$, $\Delta_{\hat{N}_o}$ can be generally addressed as uncertainties. Once they cause performance degradation, in our case the degradation in stability margin, approaching to an unacceptable level, they are called faults and the corresponding process operation is said to be faulty.

Recall that the LCF and the corresponding SKR of a plant is not unique. That means, the representation form (21.2) or (21.5) is also not unique. Suppose that $\begin{bmatrix} -\hat{N}_1 & \hat{M}_1 \end{bmatrix}$ and $\begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \end{bmatrix}$ are two different realisations of the SKR for the faulty plant. Notice that for any SKR $\begin{bmatrix} -\hat{N}_1 & \hat{M}_1 \end{bmatrix}$, there exists $R(z) \in \mathcal{RH}_\infty$ such that

$$\begin{bmatrix} -\hat{N}_2 & \hat{M}_2 \end{bmatrix} = R\begin{bmatrix} -\hat{N}_1 & \hat{M}_1 \end{bmatrix}, R^{-1}(z) \in \mathcal{RH}_\infty.$$

Hence, in a more general case, we have

$$\begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} = \begin{bmatrix} -R\hat{N} + \hat{N}_o & R\hat{M} - \hat{M}_o \end{bmatrix},$$

with $R(z)$ being any transfer matrix that belongs to $\mathcal{RH}_\infty$. It is known that the feedback control loop is stable if and only if

$$\left\| \left[-\varDelta_{\hat{N}_o} \; \varDelta_{\hat{M}_o} \right] \right\|_\infty \left\| \begin{bmatrix} -\hat{Y}_o - M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix} \right\|_\infty < 1.$$

Indeed, the different realisation forms for the SKR should not influent the stability of the process, although with different SKRs, $\left\| \left[-\varDelta_{\hat{N}_o} \; \varDelta_{\hat{M}_o} \right] \right\|_\infty$ can be different. Notice that

$$\left\| \left[-\varDelta_{\hat{N}_o} \; \varDelta_{\hat{M}_o} \right] \right\|_\infty \geq \inf_{R \in \mathcal{RH}_\infty} \left\| \left[\hat{N}_o \; -\hat{M}_o \right] - R \left[\hat{N} \; -\hat{M} \right] \right\|_\infty .$$

Thus, along the line of our discussion in Chap. 9, the influence of the uncertainty/fault on the system stability is to be uniquely represented by

$$\left[-\varDelta_{\hat{N}_o} \; \varDelta_{\hat{M}_o} \right] = \left[\hat{N}_o \; -\hat{M}_o \right] - R^* \left[\hat{N} \; -\hat{M} \right], \tag{21.6}$$

where

$$R^* = \arg \inf_{R \in \mathcal{RH}_\infty} \left\| \left[\hat{N}_o \; -\hat{M}_o \right] - R \left[\hat{N} \; -\hat{M} \right] \right\|_\infty . \tag{21.7}$$

Problem formulation

The main focus of our subsequent study is on the analysis of the performance degradation caused by $\varDelta_{\hat{M}_o}$, $\varDelta_{\hat{N}_o}$, and based on it, establishing a performance-based detection and performance degradation recovering strategy. For our purpose, we first introduce an indicator, the so-called fault-tolerant margin, to characterise the degradation of the stability margin caused by $\varDelta_{\hat{M}_o}$, $\varDelta_{\hat{N}_o}$. Notice that in the context of performance-based framework, we do not distinguish model uncertainties and the so-called faults in terms of $\varDelta_{\hat{M}_o}$, $\varDelta_{\hat{N}_o}$. Instead, normal and faulty operations will be determined by the performance degradation. To ensure the robustness against the model uncertainties and avoid false alarms, a threshold setting scheme should be thus proposed. Next, the fault-tolerant margin is estimated in the observer-based residual generation context, which is further implemented for the detection purpose. Moreover, a performance degradation recovering strategy is proposed, and associated with it, the design methodologies are investigated.

Below, the assumptions made in our subsequent work are summarised:

- the LC pair $\left(-\hat{N}_o(z), \hat{M}_o(z) \right)$ of $G(z)$ is known,
- the reference signal v satisfies the persistently excitation condition, and
- the measurements $u(k)$ and $y(k)$ are available.

21.2.2 Performance Degradation Detection and Recovering Schemes

Performance degradation detection

During process operations, the closed-loop dynamics for the feedback control system can be described by

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} I & -K \\ -G_f & I \end{bmatrix}^{-1} \begin{bmatrix} v \\ d \end{bmatrix} = \begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N} & \hat{M} \end{bmatrix}^{-1} \begin{bmatrix} \hat{V} v \\ \hat{M} d \end{bmatrix}. \tag{21.8}$$

Considering that

$$\begin{aligned} \begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N} & \hat{M} \end{bmatrix}^{-1} &= \left(\begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N}_o & \hat{M}_o \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \left(I + \begin{bmatrix} 0 & 0 \\ -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \right)^{-1}, \end{aligned}$$

it follows from small gain theorem that the closed-loop system is stable if

$$\left\| \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \right\|_\infty < 1.$$

It is clear that fault $\Delta_{\hat{N}_o}$, $\Delta_{\hat{M}_o}$ affects the system stability. For the purpose of performance degradation recovery, we introduce the following definition.

Definition 21.1 *Given the feedback control system (21.8), the value*

$$b(K) = \left\| \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \right\|_\infty \tag{21.9}$$

is called fault-tolerant margin.

It is evident that $b(K)$ is a stability performance indicator and characterises the performance degradation in the system stability. Indeed, if $b(K)$ is close to one, it indicates that the system is approaching the stability margin. It is worth noting that $b(K)$ is closely related to the loop stability margin introduced in Sect. 19.3. In fact, the higher value of the loop stability margin implies a smaller $b(K)$ and so a higher fault-tolerant margin.

In practice, the process is generally in a critical operation condition when $b(K)$ is approaching one. Let $b_{th} < 1$ be the maximum tolerance bound of the process, then the stability performance of the process can be monitored by applying the following decision logic

$$\begin{cases} b(K) < b_{th} \Longrightarrow \text{stable,} \\ b(K) \geq b_{th} \Longrightarrow \text{stability margin is approached.} \end{cases}$$

As the detection threshold, determination of b_{th} depends strongly on the system dynamics, application requirements and the applied FTC algorithm, and should be achieved by a trade-off between FAR and stability guarantee. Theoretically, when no knowledge of $\varDelta_{\hat{M}_o}$, $\varDelta_{\hat{N}_o}$ is available, it holds, for some (constant) $\delta_\varDelta$ (> 0),

$$\sup_{\left\| \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \right\|_\infty \leq \delta_\varDelta} b(K) = \delta_\varDelta \left\| \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix} \right\|_\infty .$$

So, the threshold b_{th} can be determined by varying the constant $\delta_\varDelta$ such that

$$b_{th} \leq b_{th,0},$$

where $b_{th,0}$ is the (maximal) acceptable margin of $b(K)$.

Next, we address the issue of online estimation of $b(K)$. To this end, the following residual generator is adopted

$$r(z) = -\hat{N}_o(z)u(z) + \hat{M}_o(z)y(z),$$

which can be constructed using the available model-based or data-driven methods. It is known that the dynamics of the residual generator is governed by

$$\begin{aligned} r(z) &= \varDelta_{\hat{N}_o} u(z) - \varDelta_{\hat{M}_o} y(z) + \hat{M}(z)d(z) \\ &= \begin{bmatrix} \varDelta_{\hat{N}_o} & -\varDelta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} \hat{V}(z) & \hat{U}(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix}^{-1} \begin{bmatrix} \bar{v}(z) \\ \hat{M}(z)d(z) \end{bmatrix} + \hat{M}(z)d(z), \qquad (21.10) \\ \bar{v}(z) &= \hat{V}(z)v(z). \end{aligned}$$

Note that

$$\begin{bmatrix} \hat{V} & \hat{U} \\ -\hat{N} & \hat{M} \end{bmatrix}^{-1} = \left(I + \begin{bmatrix} -U \\ V \end{bmatrix} \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \right)^{-1} \begin{bmatrix} M_o & -U \\ N_o & V \end{bmatrix}.$$

Moreover, since

$$\begin{aligned} &\begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \left(I + \begin{bmatrix} -U \\ V \end{bmatrix} \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \right)^{-1} \\ &= \left(I + \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} -U \\ V \end{bmatrix} \right)^{-1} \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix}, \end{aligned}$$

we have

$$\begin{aligned}
r(z) &= P_\Delta(z)\bar{v}(z) + \bar{d}(z), \\
P_\Delta &= -\left(I + \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} M_o \\ N_o \end{bmatrix}, \\
\bar{d} &= \left(I + \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} -U \\ V \end{bmatrix}\right)^{-1} \left(\hat{M}_o + \Delta_{\hat{M}_o}\right) d.
\end{aligned} \tag{21.11}$$

For our purpose, the relation between the fault-tolerant margin $b(K)$ and P_Δ is presented in the following theorem.

Theorem 21.1 *For $b(K) < 1$, it holds*

$$\|P_\Delta\|_\infty \leq \frac{b(K)}{\sqrt{1 - b^2(K)}}. \tag{21.12}$$

Proof Write $b(K)$ as

$$\begin{aligned}
b(K) &= \left\| \begin{bmatrix} \Delta_1 & \Delta_2 \end{bmatrix} \right\|_\infty, \\
\Delta_1 &= \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} M_o \\ N_o \end{bmatrix}, \Delta_2 = \begin{bmatrix} -\Delta_{\hat{N}_o} & \Delta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} -U \\ V \end{bmatrix}.
\end{aligned}$$

Note that

$$P_\Delta = (I + \Delta_2)^{-1} \Delta_1.$$

It follows from Lemma 9.4 that for

$$\left\| \begin{bmatrix} \Delta_1 & \Delta_2 \end{bmatrix} \right\|_\infty = b(K) < 1,$$

it holds

$$\left\| (I + \Delta_2)^{-1} \Delta_1 \right\|_\infty = \|P_\Delta\|_\infty \leq \frac{b(K)}{\sqrt{1 - b^2(K)}}.$$

The theorem is thus proved.

Inequality (21.12) implies

$$b^2(K) \geq \frac{\|P_\Delta\|_\infty^2}{1 + \|P_\Delta\|_\infty^2}. \tag{21.13}$$

That means, by means of $\|P_\Delta\|_\infty$ we are able to compute a lower bound of $b(K)$. On the assumption of $\bar{d} = 0$ and according to (21.11), $\|P_\Delta\|_\infty$ can be computed using the online data $\bar{v}$ and r. This motivates us to define

$$J(K) = \|P_\Delta\|_\infty$$

as the performance (assessment) indicator that is adopted for the assessment of the stability degradation.

$\mathcal{H}_\infty$ norm estimation of a transfer function (matrix) using data is known in robust control theory (the reader is referred to the reference given at the end of the chapter). The estimation algorithm for $J(K)$ is summarised in the following algorithm.

Algorithm 21.1 *Estimation of stability degradation indicator $J(K)$*

Step 1: *Online collecting the measurement data $r(k)$, $\bar{v}(k)$;*
Step 2: *Constructing the Hankel matrices*

$$R_{k,l,N} = \begin{bmatrix} r(k-l) & \cdots & r(k-l+N) \\ \vdots & \ddots & \vdots \\ r(k) & \cdots & r(k+N) \end{bmatrix},$$

$$\bar{V}_{k,l,N} = \begin{bmatrix} \bar{v}(k-l) & \cdots & \bar{v}(k-l+N) \\ \vdots & \ddots & \vdots \\ \bar{v}(k) & \cdots & \bar{v}(k+N) \end{bmatrix};$$

Step 3: *Recursively computing the maximal singular value*

$$J(K) = \sigma_{max}\left(R_{k,l,N}\bar{V}_{k,l,N}^T\left(\bar{V}_{k,l,N}\bar{V}_{k,l,N}^T\right)^{-1}\right). \tag{21.14}$$

Here, l and N denote sufficiently large positive integers. Generally speaking, to achieve good estimation performance, a large N is necessary, which will, in turn, result in enormous computation efforts and detection delay. From the application perspective, it is of great significance to choose a proper N to achieve a proper trade-off between the estimation performance and the computation efforts.

As the next step, a detection threshold is to be determined. Recall that b_{th} is the maximum fault-tolerant margin. It follows from (21.12) that the upper bound for $J(K)$ should be correspondingly set as

$$J_{th} = \frac{b_{th}}{\sqrt{1-b_{th}^2}}.$$

It can be seen from (21.13) that

$$J(K) \geq J_{th} \Longrightarrow b(K) \geq b_{th}.$$

Since $J(K)$ can be estimated online, the faulty operation caused by the performance degradation is detected by applying the following detection logic

$$\begin{cases} J(K) < J_{th} \Longrightarrow \text{stable}, \\ J(K) \geq J_{th} \Longrightarrow \text{stability margin is approached.} \end{cases}$$

Now, we consider the more general case for $\bar{d} \neq 0$. It is obvious that unknown input vector $\bar{d}$ will cause estimation error, as Algorithm 21.1 is applied for $\|P_{\Delta}\|_{\infty}$ estimation. In what follows, we characterise the deviation caused by $\bar{d}$. Let

$$\left\|\hat{P}_{\Delta}\right\|_{\infty} = \sigma_{max}\left(R_{k,l,N}\bar{V}_{k,l,N}^{T}\left(\bar{V}_{k,l,N}\bar{V}_{k,l,N}^{T}\right)^{-1}\right)$$

denote the estimate of $\|P_{\Delta}\|_{\infty}$ using r, $\bar{v}$. It turns out

$$\left\|\hat{P}_{\Delta}\bar{v}\right\|_{2} \approx \|r\|_{2} = \left\|P_{\Delta}\bar{v} + \bar{d}\right\|_{2},$$

which implies, in general,

$$\|P_{\Delta}\bar{v}\|_{2} - \left\|\bar{d}\right\|_{2} \leq \left\|\hat{P}_{\Delta}\bar{v}\right\|_{2} \leq \|P_{\Delta}\bar{v}\|_{2} + \left\|\bar{d}\right\|_{2}.$$

As a result, for any $\bar{v} \neq 0$, the following inequality holds

$$\frac{\|P_{\Delta}\bar{v}\|_{2}}{\|\bar{v}\|_{2}} - \frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}} \leq \frac{\left\|\hat{P}_{\Delta}\bar{v}\right\|_{2}}{\|\bar{v}\|_{2}} \leq \frac{\|P_{\Delta}\bar{v}\|_{2}}{\|\bar{v}\|_{2}} + \frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}}.$$

It yields

$$\sup_{\bar{v}\neq 0}\left(\frac{\|P_{\Delta}\bar{v}\|_{2}}{\|\bar{v}\|_{2}} - \frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}}\right) \leq \sup_{\bar{v}\neq 0}\frac{\left\|\hat{P}_{\Delta}\bar{v}\right\|_{2}}{\|\bar{v}\|_{2}} \leq \sup_{\bar{v}\neq 0}\left(\frac{\|P_{\Delta}\bar{v}\|_{2}}{\|\bar{v}\|_{2}} + \frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}}\right)$$

$$\Longrightarrow \begin{cases} \sup_{\bar{v}\neq 0}\frac{\left\|\hat{P}_{\Delta}\bar{v}\right\|_{2}}{\|\bar{v}\|_{2}} \leq \sup_{\bar{v}\neq 0}\frac{\|P_{\Delta}\bar{v}\|_{2}}{\|\bar{v}\|_{2}} + \sup_{\bar{v}\neq 0}\frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}} \\ \sup_{\bar{v}\neq 0}\frac{\left\|\hat{P}_{\Delta}\bar{v}\right\|_{2}}{\|\bar{v}\|_{2}} \geq \sup_{\bar{v}\neq 0}\frac{\|P_{\Delta}\bar{v}\|_{2}}{\|\bar{v}\|_{2}} - \inf_{\bar{v}\neq 0}\frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}} \end{cases}$$
$$\Longrightarrow \begin{cases} \left\|\hat{P}_{\Delta}\right\|_{\infty} \leq \|P_{\Delta}\|_{\infty} + \sup_{\bar{v}\neq 0}\frac{\|\bar{d}\|_{2}}{\|\bar{v}\|_{2}}, \\ \left\|\hat{P}_{\Delta}\right\|_{\infty} \geq \|P_{\Delta}\|_{\infty} - \inf_{\bar{v}\neq 0}\frac{\|\bar{d}\|_{2}}{\|\bar{v}\|_{2}} \geq \|P_{\Delta}\|_{\infty} - \sup_{\bar{v}\neq 0}\frac{\|\bar{d}\|_{2}}{\|\bar{v}\|_{2}}. \end{cases}$$

That means, the error of the estimated and the real $\|P_{\Delta}\|_{\infty}$ is bounded by

$$|J(K) - \|P_{\Delta}\|_{\infty}| \leq \sup_{\bar{v}\neq 0}\frac{\left\|\bar{d}\right\|_{2}}{\|\bar{v}\|_{2}}.$$

Definition 21.2 *The value R_{D2R} defined by*

$$R_{D2R}=\sup_{\bar{v}\neq 0}\frac{\left\|\bar{d}\right\|_2}{\left\|\bar{v}\right\|_2}$$

is called disturbance-to-reference ratio.

It is evident that the accuracy and reliability of the estimations depend on the size of the disturbance-to-reference ratio R_{D2R}. For the case of a small R_{D2R}, Algorithm 21.1 delivers a reliable estimation of $\| P_{\Delta} \|_{\infty}$ with adequate degree of accuracy.

It is to emphasise that the residual generator adopted in the proposed detection approach is standard. In spite of this, the advantages of the proposed approach, in comparison with the existing fault detection methods, lie in

- detecting/estimating the control performance degradation by using the available data in the real-time manner, and
- delivering an indicator to show whether the system is approaching the stability margin.

To our best knowledge, very limited attention has been paid on the detection and estimation issues of the control performance changes in the research domain, which are, however, of practical application interests. Our proposed approach makes a useful contribution in this thematic field.

Performance-based fault-tolerant control: a general description

It is evident that performance model $J(K)$ is a function of the controller and thus a function of the fault-tolerant margin as well. Once a fault leads to unacceptable performance degradation, a fault-tolerant controller will be applied to accommodate the performance degradation. Consider the observer-based realisation of all stabilising controllers,

$$\begin{aligned}\hat{x}(k+1)&=A\hat{x}(k)+Bu(k)+Lr(k),\\ r(k)&=y(k)-C\hat{x}(k)-Du(k),\\ u(z)&=F\hat{x}(z)+Q(z)r(z),\end{aligned}$$

with $\hat{x}(k)$ representing the state estimation. Recall that two parameters are available in this fault-tolerant control architecture for different functionalities:

- F, L, as high-priority parameters, are used to ensure the system stability, and
- $Q(z)$, as a low-priority parameter, is generally implemented for robustness and fault-tolerance purpose.

If a fault alarm is released by the detection logic, the low-priority parameter $Q(z)$ can be first plugged in/activated to recover the performance degradation without reconfiguring the operational controller. For instance, the fault-tolerant margin can be optimised by setting $Q(z)$ as

$$Q^* = \arg \inf_{Q \in \mathcal{RH}_\infty} \left\| \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} M_o & -\hat{Y}_o - M_o Q \\ N_o & \hat{X}_o - N_o Q \end{bmatrix} \right\|_\infty .$$

However, it is not always in the situation that all the degradation caused by the fault can be recovered by tuning/plugging in the lower priority parameter. That is to say, once

$$\begin{aligned} & b(K^*) \geq b_{th} \text{ or } J(K^*) \geq J_{th}, \\ & K^* = -\left(\hat{Y}_o + M_o Q^*\right)\left(\hat{X}_o - N_o Q^*\right)^{-1}, \end{aligned} \tag{21.15}$$

it is necessary to re-configure the operational controller (the high priority controller) to maintain the performance of the system. Considering in this light, we propose the following performance degradation recovering (PDR) strategy:

- if $J(K) \geq J_{th}$, the controller $Q^*(z)$ is first implemented to accommodate the performance degradation. We label this action as *PDR phase I*;
- if $J(K^*) \geq J_{th}$, the operational controller is re-constructed to recover the stability performance. We rate this scheme as *PDR phase II.*

In the sequel, we are going to address the above two *PDR* phases in the data-driven fashion.

Performance-based fault-tolerant control: PDR phase I

The core of the *PDR phase I* lies in minimising $b(K)$ by tuning $Q(z)$. This is achieved by solving the following optimisation problem

$$Q^* = \arg \inf_{Q \in RH_\infty} \left\| \begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix} \begin{bmatrix} -\hat{Y}_o - M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix} \right\|_\infty .$$

To achieve it, an algorithm for identifying $\begin{bmatrix} -\varDelta_{\hat{N}_o} & \varDelta_{\hat{M}_o} \end{bmatrix}$ is applied. It follows directly from (21.4) that this can be realised by identifying the SKR $\left(-\hat{N}, \hat{M}\right)$ for the plant with performance degradation.

In what follows, we are devoted to a recursive data-driven realisation of SKR using input/output (I/O) data. Recall the notations

$$w_l(k) = \begin{bmatrix} w(k-l) \\ \vdots \\ w(k) \end{bmatrix}, W_{k,l} = \begin{bmatrix} w_l(k) \cdots w_l(k+N-1) \end{bmatrix},$$

where l and N denote sufficiently large positive integers, and w can be any signal. Let

$$z_p = \begin{bmatrix} u_{l_p}(k+N-1) \\ y_{l_p}(k+N-1) \end{bmatrix}, Z_p = \begin{bmatrix} U_{k-1,l_p} \\ Y_{k-1,l_p} \end{bmatrix}$$

with l_p being a sufficiently large integer. It is easy to see that $\mathcal{K}_{d,l}$ is a data-driven realisation of the SKR, when it satisfies

$$\mathcal{K}_{d,l} \begin{bmatrix} U_{k+l,l} \\ Y_{k+l,l} \end{bmatrix} = 0.$$

For the identification of $\mathcal{K}_{d,l}$, the LQ-decomposition based algorithm given in Sect. 4.4 is recalled, which is summarised below.

Algorithm 21.2 *SKR identification*

Step 1: *Collect the I/O data of the system and build* $U_{k+l,l}$, $Y_{k+l,l}$, Z_p;
Step 2: *Do an LQ-decomposition*

$$\Phi = \begin{bmatrix} Z_p \\ U_{k+l,l} \\ Y_{k+l,l} \end{bmatrix} = \begin{bmatrix} L_{f,11} & 0 & 0 \\ L_{f,21} & L_{f,22} & 0 \\ L_{f,31} & L_{f,32} & L_{f,33} \end{bmatrix} \begin{bmatrix} Q_{f,1} \\ Q_{f,2} \\ Q_{f,3} \end{bmatrix};$$

Step 3: *Do an SVD of*

$$\begin{bmatrix} L_{f,21} & L_{f,22} \\ L_{f,31} & L_{f,32} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2(\approx 0) \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix};$$

Step 4: *Set* $\mathcal{K}_{d,l} = U_2^T$.

For the purpose of online update of $\mathcal{K}_{d,l}$, a recursive form of LQ decomposition can be applied. Once new measurement data is available, we have

$$\Phi_{new} = \left[\begin{array}{c|c} Z_p & z_p \\ U_{k+l,l} & u_l(k+l+N) \\ Y_{k+l,l} & y_l(k+l+N) \end{array} \right] = [\Phi |\ \phi] = L_{new} Q_{new}.$$

Recall that with Givens-transformation, L_{new} can be recursively updated by

$$[L_{new} |\ 0] = \left[\varepsilon L_f \middle|\ \phi \right] Q_{givens},$$

where $0 < \varepsilon \leq 1$ is a forgetting factor to weight the past information, and Q_{givens} is a Givens matrix. Associated with it, the data-driven realisation $\mathcal{K}_{d,l}$ can be iteratively updated.

In Sect. 4.4, a state space realisation of a parity vector has been introduced. Let $\left[\beta_l \; \alpha_l\right]$ be one row of $\mathcal{K}_{d,l}$. The state space representation of the identified SKR is given by

$$\begin{aligned} x_z(k+1) &= A_z x_z(k) + B_z u(k) + L_z y(k), \\ r_0(k) &= G y(k) - C_z x_z(k) - D_z u(k), \end{aligned} \tag{21.16}$$

where $x_z(k)$ represents the state vector for the SKR, and

$$\begin{aligned} A_z &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}, B_z = \begin{bmatrix} \beta_l(1, 1:p) \\ \beta_l(1, p+1:2p) \\ \vdots \\ \beta_l(1, (n-1)p+1:np) \end{bmatrix}, \\ C_z &= \begin{bmatrix} 0 \cdots 0 \; 1 \end{bmatrix}, D_z = \beta_l(1, np+1:(n+1)p), \\ L_z &= -\begin{bmatrix} \alpha_{l,0} \; \alpha_{l,1} \cdots \alpha_{l,l-1} \end{bmatrix}^T, G = \alpha_{l,l}. \end{aligned}$$

Let

$$\begin{aligned} r(k) &= y(k) - C_f x_z(k) - D_f u(k), \\ C_f &= G^{-1} C_z, D_f = G^{-1} D_z. \end{aligned} \tag{21.17}$$

The transfer functions for the SKR are then given by

$$\hat{M}_f = (A_z, L_z, -C_f, I), \hat{N}_f = (A_z, B_z, C_f, D_f). \tag{21.18}$$

With the SKR (21.18) at hand, once $J(K) \geq J_{th}$, $\left[-\Delta_{\hat{N}_o} \; \Delta_{\hat{M}_o}\right]$ can be estimated by dealing with the model matching problem (MMP) given in (21.6) and (21.7) online. As a result, we have

$$\begin{aligned} & R^* \begin{bmatrix} -\hat{N}_f & \hat{M}_f \end{bmatrix} \begin{bmatrix} -\hat{Y}_o - M_o Q \\ \hat{X}_o - N_o Q \end{bmatrix} - I =: \Pi_1 - \Pi_2 Q, \\ & \Pi_1 = R^*(\hat{N}_f \hat{Y}_o + \hat{M}_f \hat{X}_o) - I, \Pi_2 = R^*(\hat{N}_f M_o + \hat{M}_f N_o), \end{aligned}$$

which means in turn

$$Q^* = \arg \inf_{Q \in RH_\infty} \left\| \Pi_1 - \Pi_2 Q \right\|_\infty . \tag{21.19}$$

It yields that the fault-tolerant margin can be optimised by tuning Q in handling MMP problem (21.19). To sum up, we propose Algorithm 21.3 to realise PDR phase I.

Algorithm 21.3 *Realisation of PDR phase I*

Step 1: *If* $J(K) \geq J_{th}$*, run Algorithm* 21.2 *or the recursive algorithm for identifying* $\hat{M}_f(z), \hat{N}_f(z)$;

Step 2: *Solve* $R^*(z)$ *according to (*21.7*);*

Step 3: *Solve* $Q^*(z)$ *according to (*21.19*);*

Step 4: *Implement* $Q^*(z)r(z)$.

Performance-based fault-tolerant control: PDR phase II

We are now in a position to present an algorithm for the optimisation in the PDR phase II using process data. Remember that our task is to construct an observer-based feedback controller to recover the stability performance.

Once $J(K^*) \geq J_{th}$, based on the recursive SKR, the residual generator (21.16)-(21.17) is first constructed which delivers the state estimation $x_z(k)$ and residual signal $r(k)$ for the fault-tolerant purpose. Let us re-write the residual generator as

$$\begin{aligned} x_z(k+1) &= A_f x_z(k) + B_f u(k) + L_f r(k), \\ r(k) &= y(k) - C_f x_z(k) - D_f u(k), \end{aligned} \tag{21.20}$$

where

$$\begin{aligned} A_f &= A_z + L_f C_f, B_f = B_z + L_f D_f, \\ C_f &= G^{-1} C_z, D_f = G^{-1} D_z, \; L_f = L_z. \end{aligned}$$

As a result, the feedback controller can be given by

$$u(z) = F_f x_z(z) + v(z),$$

where F_f is the controller parameter to be determined. To this end, the following cost function is minimised with respect to F_f

$$V = \lim_{N \to \infty} \frac{1}{N} \left(x_z^T(k) W_f x_z(k) + u^T(k) R_f u(k) \right).$$

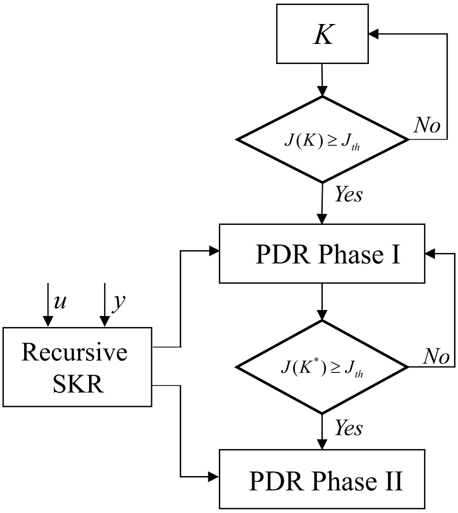
Here, $W_f \geq 0, R_f > 0$ are weighting matrices. It follows from the separation principle that the estimation and control issues can be handled independently. From the control perspective, the controller gain F_f can be determined by solving the LQR problem as

$$\begin{aligned} P &= A_f^T P A_f - A_f^T P B_f \left(B_f^T P B_f + R_f \right)^{-1} B_f^T P A_f + W_f, \\ F_f &= -\left(B_f^T P B_f + R_f \right)^{-1} B_f^T P A_f. \end{aligned} \tag{21.21}$$

The needed computations for PDR phase II are summarised in Algorithm 21.4.

Algorithm 21.4 *Realisation of PDR phase II*

Fig. 21.3 Performance degradation recovering strategy



Step 1: *If $J(K^*) \geq J_{th}$, construct the residual generator (21.20) based on the recursive SKR;*
Step 2: *Calculate F_f according to (21.21);*
Step 3: *Set the observer-based feedback controller equal to $F_f x_z(k)$.*

Finally, the schematic of the overall PDR strategy is illustrated by Fig. 21.3.

21.3 An Approach to Loop Performance Recovery

Although the approach proposed in the previous section can be applied to the recovery of loop performance degradation, as defined and discussed in Sect. 19.2, we will study in this section an alternative approach dedicated to the loop performance recovering issue. To be specific, we would like to reduce

$$\left\| \begin{matrix} e_u(k) \\ e_y(k) \end{matrix} \right\|_2 = \left\| \begin{matrix} u_{ideal}(k) - u(k) \\ y_{ideal}(k) - y(k) \end{matrix} \right\|_2$$

by tuning the parameterisation matrix $Q(z)$ of a dynamic output feedback or an observer-based state feedback controller. Here, $u_{ideal}(k)$ and $y_{ideal}(k)$ represent the ideal state feedback control signal and the corresponding plant output response. The reader is referred to Sect. 19.2 for the detail about the definitions and formulation related to the loop performance degradation.

The basic idea behind the approach is to model and identify the faulty part in the process which causes loop performance degradation, and to manage performance recovery by tuning the controller parameter $Q(z)$.

21.3.1 Dual Form of Youla Parameterisation and Parameterisation of Process Uncertainty

In Sect. 5.2, we have described the so-called parameterisation of Bezout-Identity. In fact, this is the mathematical form of the so-called dual form of the Youla parameterisation. To be specific, consider a feedback control loop with the plant model

$$\begin{aligned} y(z) &= G(z)u(z), \\ G(z) &= C(zI-A)^{-1}B+D = N_o(z)M_o^{-1}(z) = \hat{M}_o^{-1}(z)\hat{N}_o(z), \end{aligned}$$

and the feedback controller

$$u(z) = K(z)y(z), K(z) = -\hat{Y}_o(z)\hat{X}_o^{-1}(z) = -X_o^{-1}(z)Y_o(z), \tag{21.22}$$

as described in the previous section. Here, (M_o, N_o), $\left(\hat{M}_o, \hat{N}_o\right)$ are the RC and LC pairs of $G(z)$, and (X_o, Y_o), $\left(\hat{X}_o, \hat{Y}_o\right)$ the RC and LC pairs of $K(z)$, respectively. These eight transfer matrices form the Bezout identity. It is imaginable that the controller (21.22) is able to stabilise a number of plants. The dual form of the Youla parameterisation, well-known in robust control theory (see the reference given at the end of this chapter), provides us with a parameterisation of all those plant models, which can be stabilised by the controller (21.22). It is described by

$$\begin{aligned} G_f(z) &= \left(N_o(z) - \hat{X}_o(z)S(z)\right)\left(M_o(z) + \hat{Y}_o(z)S(z)\right)^{-1} \\ &= \left(\hat{M}_o(z) + S(z)Y_o(z)\right)^{-1}\left(\hat{N}_o(z) - S(z)X_o(z)\right) \end{aligned} \tag{21.23}$$

with the parameterisation matrix $S(z) \in \mathcal{RH}_\infty$. In the context of FTC (robust control as well) $S(z)$ is adopted to model uncertainties that lead to control performance degradation.

In order to get a deeper understanding of $S(z)$ and, associated with it, some important facts, we recall our discussion on $X_o(z)$, $Y_o(z)$ in Sect. 5.2. A state space realisation of

$$\begin{bmatrix} X_o(z) & Y_o(z) \end{bmatrix} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix}$$

is the observer-based state feedback control,

$$\begin{aligned} \hat{x}(k+1) &= (A-LC)\,\hat{x}(k) + (B-LD)\,u(k) + Ly(k), \\ u(k) &= F\hat{x}(k) + v(k) \Longleftrightarrow v(k) = u(k) - F\hat{x}(k) \\ &\Longrightarrow \begin{bmatrix} X_o(z) & Y_o(z) \end{bmatrix} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = v(z). \end{aligned}$$

Remark 21.1 *Signal vector v may consist of the (known) reference signal and some unknown inputs representing uncertainties, as demonstrated in the sequel.*

It is straightforward from (21.23) that

$$\begin{aligned}
&y(z) = G_f(z)u(z) \Longrightarrow \\
&\left(\hat{M}_o(z) + S(z)Y_o(z)\right) y(z) = \left(\hat{N}_o(z) - S(z)X_o(z)\right) u(z) \Longrightarrow \\
&\hat{M}_o(z)y(z) - \hat{N}_o(z)u(z) = r(z) = -S(z)v(z).
\end{aligned} \tag{21.24}$$

As expected, the residual vector r is driven by v due to the existence of the uncertainty $S(z)$.

Now, we consider a controller in the general form

$$\begin{aligned}
&K(z) = -U(z)V^{-1}(z) = -\hat{V}^{-1}(z)\,\hat{U}(z), \\
&\begin{bmatrix} \hat{V} & \hat{U} \end{bmatrix} = \begin{bmatrix} X_o - Q\hat{N}_o & Y_o + Q\hat{M}_o \end{bmatrix}, \begin{bmatrix} V \\ U \end{bmatrix} = \begin{bmatrix} \hat{X}_o - N_o Q \\ \hat{Y}_o + M_o Q \end{bmatrix}.
\end{aligned}$$

It is evident that the uncertainty $S(z)$ will affect system stability. To illustrate it, we check the condition, under which

$$\begin{bmatrix} I & -K(z) \\ -G_f(z) & I \end{bmatrix}^{-1} \in \mathcal{RH}_\infty.$$

From the following computations,

$$\begin{aligned}
\begin{bmatrix} I & -K \\ -G_f & I \end{bmatrix}^{-1} &= \begin{bmatrix} X_o - Q\hat{N}_o & Y_o + Q\hat{M}_o \\ SX_o - \hat{N}_o & \hat{M}_o + SY_o \end{bmatrix}^{-1} \begin{bmatrix} X_o - Q\hat{N}_o & 0 \\ 0 & \hat{M}_o + SY_o \end{bmatrix}, \\
\begin{bmatrix} X_o - Q\hat{N}_o & Y_o + Q\hat{M}_o \\ SX_o - \hat{N}_o & \hat{M}_o + SY_o \end{bmatrix} &= \begin{bmatrix} I & Q \\ S & I \end{bmatrix} \begin{bmatrix} X_o & Y_o \\ -\hat{N}_o & \hat{M}_o \end{bmatrix}, \\
\begin{bmatrix} X_o & Y_o \\ -\hat{N}_o & \hat{M}_o \end{bmatrix}^{-1} &= \begin{bmatrix} M_o & -\hat{Y}_o \\ N_o & \hat{X}_o \end{bmatrix},
\end{aligned}$$

it becomes clear that the closed-loop system is stable if and only if

$$\begin{bmatrix} I & Q \\ S & I \end{bmatrix}^{-1} \in \mathcal{RH}_\infty. \tag{21.25}$$

It is remarkable that, for the given controller

$$u(z) = -\left(X_o(z) - Q(z)\hat{N}_o(z)\right)^{-1} \left(Y_o(z) + Q(z)\hat{M}_o(z)\right) y(z),$$

it holds

$$\begin{aligned}\left(X_o(z) - Q(z)\hat{N}_o(z)\right)u(z) = -\left(Y_o(z) + Q(z)\hat{M}_o(z)\right)y(z) \Longrightarrow\\ v(z) = -Q(z)r(z).\end{aligned} \tag{21.26}$$

Relations (21.24) and (21.26) imply a closed-loop with $(-Q, -S)$ as controller-plant pair and (r, v) as input–output signal pair. The stability condition of this feedback loop is given by (21.25).

21.3.2 Loop Performance Degradation Model and Problem Formulation

We now integrate the relations (21.24) and (21.26) into the LPDM established in Sect. 19.3, in order to model the loop performance degradation caused by $S(z)$. It yields

$$e_{LPD}(z) = \begin{bmatrix} e_u(z) \\ e_y(z) \end{bmatrix} = \begin{bmatrix} \hat{Y}_o(z) + M_o(z)Q(z) \\ -\hat{X}_o(z) + N_o(z)Q(z) \end{bmatrix} r(z) \tag{21.27}$$

$$= \begin{bmatrix} \hat{Y}_o(z) & -M_o(z) \\ -\hat{X}_o(z) & -N_o(z) \end{bmatrix} \begin{bmatrix} r(z) \\ v(z) \end{bmatrix}. \tag{21.28}$$

In addition, noise $\bar{w}(k)$ in the actuator modelled as

$$u(z) = K(z)y(z) + \bar{w}(z) = u_0(z) + \bar{w}(z),$$

and further measurement noise vector $\bar{\eta}(k)$,

$$y(z) = G_f(z)u(z) + \bar{\eta}(z),$$

are taken into account. We assume that, for the residual generation and state estimation purpose, only $u_0(z)$ and $y(z)$ are available and used. That means in turn,

$$\begin{aligned} r(z) &= \hat{M}_o(z)y(z) - \hat{N}_o(z)u_0(z), \\ v(z) &= Y_o(z)y(z) + X_o(z)u_0(z) \end{aligned}$$

can be computed and are thus available. Notice the relations,

$$\begin{aligned} y(z) =& G_f(z)u(z) + \bar{\eta}(z) \Longleftrightarrow \\ & \left(\hat{M}_o(z) + S(z)Y_o(z)\right)(y(z) - \bar{\eta}(z)) = \left(\hat{N}_o(z) - S(z)X_o(z)\right)u(z) \Longrightarrow \\ r(z) =& - S(z)\left(v(z) + X_o(z)\bar{w}(z) - Y_o(z)\bar{\eta}(z)\right) + \hat{M}_o(z)\bar{\eta}(z) + \hat{N}_o(z)\bar{w}(z), \end{aligned}$$

and write the last one as

$$r(z) = -S(z)\left(v(z) + w(z)\right) + \eta(z), \tag{21.29}$$

$$\begin{bmatrix} w(z) \\ \eta(z) \end{bmatrix} = \begin{bmatrix} X_o(z) & -Y_o(z) \\ \hat{N}_o(z) & \hat{M}_o(z) \end{bmatrix} \begin{bmatrix} \bar{w}(z) \\ \bar{\eta}(z) \end{bmatrix}. \tag{21.30}$$

Finally, we have the new LPDM,

$$\begin{aligned} \begin{bmatrix} r(z) \\ v(z) \end{bmatrix} &= \begin{bmatrix} I & S(z) \\ Q(z) & I \end{bmatrix}^{-1} \begin{bmatrix} -S(z)w(z) + \eta(z) \\ 0 \end{bmatrix} \Longrightarrow \\ \begin{bmatrix} e_u(z) \\ e_y(z) \end{bmatrix} &= \begin{bmatrix} \hat{Y}_o(z) & -M_o(z) \\ -\hat{X}_o(z) & -N_o(z) \end{bmatrix} \begin{bmatrix} I & S(z) \\ Q(z) & I \end{bmatrix}^{-1} \begin{bmatrix} \vartheta(z) \\ 0 \end{bmatrix}, \\ \vartheta(z) &= -S(z)w(z) + \eta(z) = \begin{bmatrix} -S(z) & I \end{bmatrix} \begin{bmatrix} X_o(z) & -Y_o(z) \\ \hat{N}_o(z) & \hat{M}_o(z) \end{bmatrix} \begin{bmatrix} \bar{w}(z) \\ \bar{\eta}(z) \end{bmatrix}. \end{aligned} \tag{21.31}$$

Since (M_o, N_o) and $\left(\hat{X}_o, \hat{Y}_o\right)$ are fixed, once the nominal system model and the state feedback gain F are given, reducing $e_{LPD}(z)$ or recovering the loop performance is achievable by tuning $Q(z)$ so that the norm of the signals,

$$\begin{bmatrix} r(z) \\ v(z) \end{bmatrix} = \begin{bmatrix} I & S(z) \\ Q(z) & I \end{bmatrix}^{-1} \begin{bmatrix} \vartheta(z) \\ 0 \end{bmatrix} \Longleftrightarrow \tag{21.32}$$

$$r(z) = -S(z)\left(v(z) + w(z)\right) + \eta(z),\, v(z) = -Q(z)r(z), \tag{21.33}$$

decreases. In this way, the original loop performance recovering problem is transformed into change detection and optimal control in the feedback control loop (21.32), as schematically sketc.hed in Fig. 21.4.

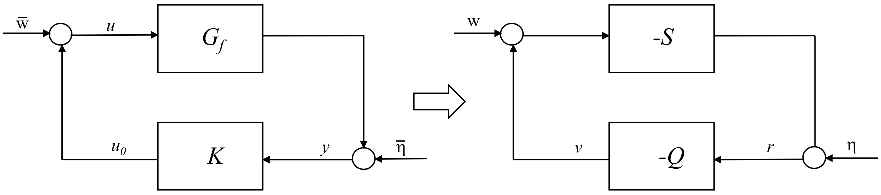


Fig. 21.4 Schematic description of problem re-formulation

In the previous chapters, for instance in Chap. 9 or Chap. 19, numerous approaches and algorithms have been introduced to deal with fault or change detection and control issues for the feedback control loop (21.32). Below, we propose an alternative approach. It consists of two steps:

- performance monitoring and loop performance degradation detection, and
- recovering algorithms.

21.3.3 Loop Performance Monitoring and Degradation Detection

To simplify our study, we assume that

$$\bar{w}(k) = 0, \bar{\eta}(k) \sim \mathcal{N}\left(0, \Sigma_{\bar{\eta}}\right),$$

and the observer gain matrix L adopted for building $\left(\hat{M}_o, \hat{N}_o\right)$ is the Kalman filter gain that delivers the white residual vector with minimum covariance in case of the existence of noise $\bar{\eta}(z)$. Note that the state space realisation of $\eta(z) = \hat{M}_o(z)\bar{\eta}(z)$ is given by

$$\begin{aligned} x_{\bar{\eta}}(k+1) &= (A - LC)\, x_{\bar{\eta}}(k) + L\bar{\eta}(k), \\ \eta(k) &= \bar{\eta}(k) - Cx_{\bar{\eta}}(k), \end{aligned}$$

which is obviously the dynamics of the Kalman filter based residual generator. In other words, $\eta(k)$ is white and satisfies

$$\eta(k) \sim \mathcal{N}\left(0, \Sigma_{\eta}\right).$$

In order to handle the color noise vector $w(z) = -Y_o(z)\bar{\eta}(z)$, we extend the subsystem $S(z)$ to

$$\begin{aligned} &\bar{S}(z) = S(z)\begin{bmatrix} I & -Y_o(z) \end{bmatrix} \Longrightarrow \\ &- S(z)\left(v(z) + w(z)\right) = -S(z)\left(v(z) - Y_o(z)\bar{\eta}(z)\right) = -\bar{S}(z)\begin{bmatrix} v(z) \\ \bar{\eta}(z) \end{bmatrix}. \end{aligned}$$

As a result, the overall system dynamics is described by

$$r(z) = -\bar{S}(z)\begin{bmatrix} v(z) \\ \bar{\eta}(z) \end{bmatrix} + \eta(z), v(z) = -Q(z)r(z) \tag{21.34}$$

with measurement vectors $r(z)$, $v(z)$, and driven by the white noises $\eta(z)$, $\bar{\eta}(z)$.

Remark 21.2 *The assumptions on $\bar{w}(k)$ and L does not lead to the loss of generality. Analogue to the above handling of color noises, the more general case (without assumptions) can be addressed by extending the uncertain system $S(z)$ so that the overall system is driven by white noises $\bar{\eta}(k)$, $\bar{w}(k)$ with measurements $r(k)$, $v(k)$.*

We are now in the position to present the monitoring scheme. Denote the state space realisation of the feedback control loop (21.34) by

$$x_{CL}(k+1) = A_{CL}x_{CL}(k) + B_{CL}\theta(k), x_{CL}(k) = \begin{bmatrix} x_Q(k) \\ x_S(k) \end{bmatrix}, \tag{21.35}$$

$$\begin{bmatrix} r(k) \\ v(k) \end{bmatrix} = C_{CL}x_{CL}(k) + D_{CL}\theta(k), \tag{21.36}$$

$$\theta(k) = \begin{bmatrix} \eta(k) \\ \bar{\eta}(k) \end{bmatrix} \sim \mathcal{N}(0, \Sigma_\theta), \tag{21.37}$$

where $x_{CL}(k)$ represents the state vector of the control loop consisting of the state vectors of systems $Q(z)$ and $-\bar{S}(z)$, $x_Q(k)$ and $x_S(k)$, respectively, and $\theta(k)$ is white, uncorrelated with $x_{CL}(k)$. Since the closed-loop is stable, the value of the performance function,

$$J(i) = \mathcal{E}\sum_{k=i}^{\infty} \gamma^{k-i}\left(r^T(k)r(k) + v^T(k)v(k)\right), 0 < \gamma < 1, \tag{21.38}$$

is given by, as shown in Chap. 19–20,

$$J(i) = x_{CL}^T(i)Px_{CL}(i) + c, \tag{21.39}$$

where $P \geq 0$ is the solution of the following Lyapunov equation,

$$P = \gamma A_{CL}^T P A_{CL} + C_{CL}^T C_{CL},$$

and c is a constant satisfying

$$c = \frac{tr\left(\left(D_{CL}^T D_{CL} + \gamma B_{CL}^T P B_{CL}\right)\right)\Sigma_\theta}{1-\gamma}.$$

Equation (21.39) is the online assessment and prediction model for the loop performance degradation and builds the basis for LPD monitoring and detection. Since the overall system (21.35)–(21.36) is unknown and thus P, c in model (21.39) are to be determined. The idea behind LPD monitoring and detection based on (21.39) is to identify P, c using the collected process data.

Recall that $x_{CL}(k)$ consists of $x_Q(k)$ and $x_S(k)$, where $x_Q(k)$ as the state vector of the known system $Q(z)$ is available, while $x_S(k)$ is, as the state vector of $-\bar{S}(z)$, unknown. On the other hand, we have demonstrated in Sect. 4.4.1 that state vector can be well estimated using process input and output data. Applying this result to our case leads to

$$\hat{x}_S(k) = L_p \begin{bmatrix} v_s(k) \\ r_s(k) \end{bmatrix}, L_p \in \mathcal{R}^{n \times (m+p)(s+1)}, \tag{21.40}$$

with some sufficiently large integer s. Substituting $x_S(k)$ by $\hat{x}_S(k)$ in (21.40) results in finally an approximation model for the online prediction of LPD value,

$$J(i) = x_{CL}^T(i) P x_{CL}(i) + c \approx \zeta^T(i) P_\zeta \zeta(i) + c, \tag{21.41}$$
$$\zeta(i) = \begin{bmatrix} x_Q(i) \\ v_s(i) \\ r_s(i) \end{bmatrix}, P_\zeta = \begin{bmatrix} I & 0 \\ 0 & L_p^T \end{bmatrix} P \begin{bmatrix} I & 0 \\ 0 & L_p \end{bmatrix}.$$

For the online identification of P_ζ, c, the approach proposed in Sect. 20.3.2 is applied. We summarise the major computation steps as follows without detailed discussion.

Write (21.41) into

$$J(i) = \zeta^T(i) P_\zeta \zeta(i) + c = \omega^T \phi(i), \tag{21.42}$$

where $\omega \in \mathcal{R}^\eta$,

$$\eta = \left(n_q + (m+p)(s+1) + 1\right)\left(n_q + (m+p)(s+1)\right)/2 + 1,$$

is the parameter vector including all parameters to be identified with n_q as the order of system $Q(z)$, and $\phi(i)$ is the corresponding vector of time functions consisting of the available process data. Considering that

$$J(i) = r^T(i) r(i) + v^T(i) v(i) + \gamma J(i+1),$$

and substituting $J(i)$, $J(i+1)$ by $\omega^T \phi(i)$, $\omega^T \phi(i+1)$, we have finally

$$\omega^T \left(\phi(i) - \gamma \phi(i+1)\right) = r^T(i) r(i) + v^T(i) v(i). \tag{21.43}$$

Model (21.43) allows us to identify ω, and, based on it, to compute $J(i)$. Thus, by a given threshold J_{th}, and detection logic,

$$J(i) > J_{th} \Longrightarrow \textit{alarm and activating recovery algorithm,}$$

LPD monitoring and detection is achieved. For the online implementation of the above scheme, it is recommended to run Algorithm 20.1.

21.3.4 Loop Performance Degradation Recovery

Concerning the loop performance degradation recovery by tuning $Q(z)$, different schemes can be applied. One possibility is to

- identify $-\bar{S}(z)$ using control loop data $v(k), r(k)$, and, based on it,
- online optimise $Q(z)$.

For the realisation of the above two steps, algorithms developed in the previous chapters can be applied. We summarise them into the following algorithm.

Algorithm 21.5 *Loop performance degradation recovery*

Step 1: *Run Algorithm 21.2 or the recursive algorithm for identifying the SKR of $-\bar{S}(z)$ using control loop data $v(k), r(k)$;*

Step 2: *Set the state space model of $-\bar{S}(z)$ equal to*

$$\begin{aligned} x_S(k+1) &= A_S x_S(k) + B_S v(k) + L_S r(k), \\ r(k) &= C_S x_S(k) + D_S v(k), \end{aligned}$$

where the system matrices are as defined in (21.16);

Step 3: *Run a Kalman filter based on the above model;*

Step 4: *Run an LQG controller as the optimal $Q(z)$,*

$$Q : \begin{cases} \hat{x}_S(k+1) = (A_S - L_S C_S)\,\hat{x}_S(k) + (B_S - L_S D_S)\,v(k) \\ \qquad\qquad +L_K \left(r(k) - C_S \hat{x}_S(k) - D_S v(k)\right), \\ v(k) = F\hat{x}_S(k), \end{cases}$$

where L_K, F are the Kalman filter gain and LQ gain matrices, respectively.

Alternatively, we can also apply the data-driven algorithms proposed in the next chapter for our purpose of loop performance degradation recovery.

21.4 Notes and References

Fault-tolerant control has become, without any doubt, one of the vital thematic areas in control theory in recent years. The great number of publications on FTC methods verify this development. Representatively, we would like to mention the monographs [1–3] and the survey papers [4–6]. It can be observed that major application and research attention has been paid to the active FTC strategies in recent years, see for instance [7–11]. Moreover, it is the state of the art in the FTC research that the existing schemes and methods are component oriented FTC. That is, they are mainly dedicated to the development of fault-tolerant control algorithms with respect to faulty system components like sensors, actuators and some other hardware components. In this context, most of the existing FTC schemes follow the strategy of compensating the influence of the faulty component on the system dynamics.

We would like to call the reader's attention to the considerable efforts in automation industry to increase the component reliability and to the trend of enhancing the intelligent degree of those key system components. Nowadays, smart sensors and actuators are state of the art in many industrial sectors, and the new generation of smart system components are of the ability of self-diagnosis and self-repair. Considering this industrial development, our research focus should be on FTC at the system level.

In this chapter, we have proposed to deal with FTC issues at the system performance level. Performance-based FTC is an active FTC strategy. In its early development, this class of FTC schemes has mainly focused on the real-time performance optimisation, in order to recover the control performance degradation [12–14], in which standard fault diagnosis schemes, typically observer-based ones, have been applied to triggering the FTC algorithm. This is also the major difference to the performance-based FTC strategy investigated in this chapter. As illustrated in Fig. 21.1, a performance-based fault detection builds the basis for our FTC schemes, which has a double role: serving (i) as a process monitoring sub-system and (ii) as a performance evaluator embedded in the FTC algorithm. It should be emphasised that, as a consequence of applying performance-based fault detection strategy, we do not strictly distinguish between model uncertainties and faults. We consider all those changes in the system as uncertainties as far as they only cause moderate performance degradation, and assess system operations as faulty, when the system performance degradation reaches an unacceptable level.

In this chapter, we have introduced two performance-based FTC approaches. The first one is dedicated to recovering the degradation in the system stability margin. In fact, this approach is developed on the basis of our study in Sect. 19.3, in which the relation between the system stability margin and the SIR of the controller plays an essential role. For our purpose, we have

- defined the fault-tolerant margin $b(K)$ as an indicator for the stability degradation,
- derived a lower bound of $b(K)$, $J(K)$, and adopted it for assessing the stability degradation.

Moreover, we have

- proposed an algorithm for the online computation of $J(K)$, in which $\mathcal{H}_\infty$ norm estimation of a transfer function using data is to be performed. To this end, we have applied the method described in [15].

In this manner, an online monitoring of the control performance (here the stability margin) is realised. For the performance recovering objective, a two-phase procedure has been proposed, in which

- the parameterisation matrix $Q(z)$ is updated to accommodate the performance degradation in the PDR phase I, and

- in the PDR phase II, an SKR identification of the plant is first performed and, based on it, the controller is optimised.

It is noteworthy that thanks to the relations between the SIR of the controller and stability margin, loop performance degradation and fault detectability, as revealed in Sect. 19.3, the proposed approach can also be applied to recovering loop performance degradations as well as to increasing fault detectability. Recently, this FTC strategy has been extended to singular systems [16] and general nonlinear systems [17]. It has also been successfully applied to the laboratory three-tank system [18].

The second approach is an alternative scheme to the first approach towards recovering loop performance degradation. Based on the well-established dual form of Youla parameterisation [19], the problem of recovering loop performance degradation is transformed into a feedback control problem with (Q, S) as the system pair of the control loop and (r, v) as control output–input signal pair. This allows us

- to model the loop performance degradation as a quadratic cost function of the control output–input signal pair (r, v) , and based on it,
- to monitor and to predict loop performance degradation using the algorithms proposed in Sect. 20.3.2.

Concerning recovering loop performance degradation by tuning the parameterisation matrix $Q(z)$, we can apply either the algorithms proposed in the first approach or the data-driven methods to be presented in the subsequent chapter.

At the end of our discussion, we would like to point out that successful FTC and performance degradation recovery often presuppose sufficient configurability of the control system under consideration [20]. This issue has not be addressed in our work, although it is of considerable practical importance. We refer the reader to the survey paper by Wang et al. [20] for a systematic and excellent review and investigation on this topic.

References

1. M. Mahmoud, J. Jiang, and Y. Zhang, *Active Fault Tolerant Control Systems*. London: Springer, 2003.
2. M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control, 2nd Edition*. Berlin Heidelberg: Springer, 2006.
3. H. Noura, D. Theilliol, J. Ponsart, and A. Chamseddine, *Faul-Tolerant Control Systems: Design and Practical Applications*. New York, NY, USA: Springer, 2009.
4. Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Review in Control*, vol. 32, pp. 229–252, 2008.
5. I. Hwang, S. Kim, Y. Kim, and C. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Trans. Contr. Syst. Tech.*, vol. 18, pp. 636–653, 2010.
6. S. Yin, B. Xiao, S. X. Ding, and D. Zhou, "A review on recent development of spacecraft attitude fault-tolerant control system," *IEEE Trans. on Industrial Electronics*, vol. 63, pp. 3311–3320, 2016.
7. K. Zhou and Z. Ren, "A new controller architecture for high performance, robust, and fault-tolerant control," *IEEE Trans. on Autom. Contr.*, vol. 46, pp. 1613–1618, 2001.

8. B. Jiang, Z. Gao, P. Shi, and Y. Xu, "Adaptive fault-tolerant tracking control of near-space vehicle using Takagi-Sugeno fuzzy models," *IEEE Trans. on Fuzzy Syst*, vol. 18, pp. 1000–1007, 2010.
9. X. Zhang, M. M. Polycarpou, and T. Parisini, "Adaptive fault diagnosis and fault-tolerant control of MIMO nonlinear uncertain systems," *Int. J. of Contr.*, vol. 83, pp. 1054–1080, 2010.
10. M. Liu and P. Shi, "Sensor fault estimation and tolerant control for ito stochatic systems with a descriptor sliding mode approach," *Automatica*, vol. 49, pp. 1242–1250, 2013.
11. Y. Yang, L. Li, and S. X. Ding, "A control-theoretic study on Runge-Kutta methods with application to real-time fault-tolerant control of nonlinear systems," *IEEE Trans. on Industrial Electronics*, vol. 62, pp. 3914–3922, 2015.
12. Y. Yang, Y. Zhang, S. X. Ding, and L. Li, "Design and implementation of lifecycle management for industrial control applications," *IEEE Trans. on Control Systems Technology*, vol. 23, pp. 1399–1410, 2015.
13. S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. on Industrial Electronics*, vol. 61, pp. 2402–2411, 2014.
14. H. Luo, X. Yang, M. Kruger, S. X. Ding, and K. Peng, "A plug-and play monitoring and control architecture for disturbance compensation in rolling mills," *IEEE-ASME Trans. on Mechatronics*, vol. 23, pp. 200–210, 2018.
15. K. Zhou, *Essential of Robust Control*. Englewood Cliffs, NJ: Prentice-Hall, 1998.
16. D. Liu, Y. Yang, L. Li, and S. X. Ding, "Control performance-based fault-tolerant control strategy for singular systems," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. (Early Access), 2020.
17. H. Han, Y. Yang, L. Li, and S. X. Ding, "Performance-based fault detection and fault-tolerant control for nonlinear systems with t-s fuzzy implementation," *IEEE Trans. on Cybernetics*, vol. (Early Access), 2020.
18. L. Li, H. Luo, S. X. Ding, Y. Yang, and K. Peng, "Performance-based fault detection and fault-tolerant control for automatic control systems," *Automatica*, vol. 99, pp. 308–316, 2019.
19. T.-T. Tay, I. Mareels, and J. B. Moore, *High Performance Control*. Springer Science + Business Media, 1998.
20. D.-Y. Wang, Y.-Y. Tu, C.-R. Liu, Y.-Z. He, and W.-B. Li, "Conotation and research of reconfigurability for space control systems: A review," *Acta Automatica Sinica*, vol. 43, pp. 1687–1702, 2017.

Chapter 22
Data-Driven Fault-Tolerant Control Schemes

In the previous chapters, fault-tolerant control and performance degradation recovering issues have been addressed mainly in the model-based fashion. Even so, identification of data-driven SIR and SKR models is often embedded in an FTC algorithm, as for instance adopted in Sect. 21.2. This motivates us to study issues of closed-loop identification of data-driven SKR and SIR in the first part of this chapter. On the basis of data-driven SKR and SIR models, we will then investigate data-driven FTC issues. The objective of this work is to deal with such a scenario often met in industrial applications: the system performance degrades to a level and an additional controller should be added to recover the performance reaching a satisfactory level.

22.1 Closed-Loop Identification of Data-Driven SIR and SKR

We begin with the identification of data-driven SIR and SKR of feedback control systems.

22.1.1 Data-Driven SIR, SKR, and Problem Formulation

We consider (internally) stable feedback control loops modelled by

$$y(z) = G(z)u(z) + \eta(z), \; y \in \mathcal{R}^m, u \in \mathcal{R}^p, \tag{22.1}$$

$$u(z) = K(z)y(z) + w(z), \tag{22.2}$$

$$K(z) = -U(z)V^{-1}(z) = -\hat{V}^{-1}(z)\hat{U}(z), \tag{22.3}$$

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5_22

and suppose that

$$G = (A, B, C, D), A \in \mathcal{R}^{n \times n}, B \in \mathcal{R}^{n \times p}, C \in \mathcal{R}^{m \times n}, D \in \mathcal{R}^{m \times p},$$

is the minimal state space realisation, $\eta(k) \sim \mathcal{N}\left(0, \Sigma_\eta\right)$ is white noise vector, $\left(\hat{V}, \hat{U}\right)$ and (V, U) are the LC and RC pair of the controller $K(z)$, respectively, and $w(z)$ is the reference vector. Recall the definitions of SIR and SKR for the plant model $G(z)$,

$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \upsilon(z),$$
$$r(z) = \begin{bmatrix} -\hat{N}(z) & \hat{M}(z) \end{bmatrix} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix}$$

with $\upsilon(z)$ representing some l_2-bounded signal and $r(z)$ the residual vector. Recently, definitions of data-driven SIR and SKR have been introduced in the literature. It is the first task of this section to define data-driven SIR and SKR for the feedback control loops given by (22.1)–(22.2).

We begin with the observer-based input–output model introduced in Sect. 19.1,

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr(k), r(k) = y(k) - \hat{y}(k), \tag{22.4}$$
$$y(k) = r(k) + C\hat{x}(k) + Du(k), \tag{22.5}$$

and substitute the controller by

$$u(z) = K(z)y(z) + w(z) = F\hat{x}(z) - Q(z)r(z) + \hat{V}(z)w(z),$$

which yields

$$\hat{x}(k+1) = (A + BF)\hat{x}(k) + B\bar{w}(k) + r_1(k), \tag{22.6}$$
$$u(k) = F\hat{x}(k) + r_2(k) + \bar{w}(k), \tag{22.7}$$
$$y(k) = r_3(k) + (C + DF)\hat{x}(k) + D\bar{w}(k), \tag{22.8}$$
$$\bar{w}(z) = \hat{V}(z)w(z), \begin{bmatrix} r_1(z) \\ r_2(z) \\ r_3(z) \end{bmatrix} = \begin{bmatrix} L - BQ(z) \\ -Q(z) \\ I - DQ(z) \end{bmatrix} r(z).$$

In the above loop model, L is the observer gain matrix adopted in the observer-based realisation of Youla parameterised feedback controller, and r is the corresponding residual signal. $r_1(k)$, $r_2(k)$ and $r_3(k)$ are color noises, and in the steady state

$$r_1(k) \sim \mathcal{N}\left(0, \Sigma_{r_1}\right), \ r_2(k) \sim \mathcal{N}\left(0, \Sigma_{r_2}\right), \ r_3(k) \sim \mathcal{N}\left(0, \Sigma_{r_3}\right).$$

It is evident that the (model-based) SIR of the above loop is

$$\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} \bar{w}(z).$$

Remember that the loop is stable and, for a large integer s,

$$A_F^s \approx 0, A_F = A + BF.$$

Hence, the loop dynamics (22.6)–(22.8) can be well approximated by means of the following input–output model:

$$\begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} = \begin{bmatrix} \Gamma_{u,s} \\ \Gamma_{y,s} \end{bmatrix} L_p \bar{w}_{s-1}(k-s-1) + \begin{bmatrix} H_{u,\bar{w},s} \\ H_{y,\bar{w},s} \end{bmatrix} \bar{w}_s(k) + \begin{bmatrix} \alpha_{u,s}(k) \\ \alpha_{y,s}(k) \end{bmatrix}, \quad (22.9)$$

$$\begin{bmatrix} \alpha_{u,s}(k) \\ \alpha_{y,s}(k) \end{bmatrix} = \begin{bmatrix} \Gamma_{u,s} \\ \Gamma_{y,s} \end{bmatrix} L_\alpha r_{1,s-1}(k-s-1) + \begin{bmatrix} H_{u,r,s} \\ H_{y,r,s} \end{bmatrix} r_{1,s}(k) + \begin{bmatrix} r_{2,s}(k) \\ r_{3,s}(k) \end{bmatrix},$$

$$\Gamma_{y,s} = \begin{bmatrix} C_F \\ C_F A_F \\ \vdots \\ C_F A_F^s \end{bmatrix} \in \mathcal{R}^{(s+1)m \times n}, \Gamma_{u,s} = \begin{bmatrix} F \\ F A_F \\ \vdots \\ F A_F^s \end{bmatrix} \in \mathcal{R}^{(s+1)p \times n},$$

$$C_F = C + DF, L_p = \begin{bmatrix} A_F^{s-1} B \cdots B \end{bmatrix}, L_\alpha = \begin{bmatrix} A_F^{s-1} \cdots I \end{bmatrix},$$

$$H_{u,\bar{w},s} = \begin{bmatrix} I & 0 & & \\ FB & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ FA_F^{s-1}B & \cdots & FB & I \end{bmatrix}, H_{y,\bar{w},s} = \begin{bmatrix} D & 0 & & \\ C_F B & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ C_F A_F^{s-1} B & \cdots & C_F B & D \end{bmatrix},$$

$$H_{u,r,s} = \begin{bmatrix} 0 & 0 & & \\ F & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ FA_F^{s-1} & \cdots & F & 0 \end{bmatrix}, H_{y,r,s} = \begin{bmatrix} 0 & 0 & & \\ C_F & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 0 \\ C_F A_F^{s-1} & \cdots & C_F & 0 \end{bmatrix}.$$

Remark 22.1 *The definition of the notations $u_s(k)$, $y_s(k)$, $\bar{w}_s(k)$ is consistent with $\omega_s(k)$ defined in Sub-section 4.4.1. That is, $\omega_s(k)$ is a column vector composed of $\omega(j)$, $j \in [k-s, k]$. Similar to it, $\omega_{\beta,s}(k)$ is adopted to denote*

$$\omega_{\beta,s}(k) = \begin{bmatrix} \omega_\beta(k-s) \\ \vdots \\ \omega_\beta(k) \end{bmatrix}.$$

Here, β could be an alphabetic character or a number. In our subsequent work, we will consistently adopt these notations.

Remark 22.2 *In the model (22.9), it is generally assumed that $s \geq n$, in order to achieve a good approximation of the system dynamics. In the data-driven framework, a sufficiently large s is often selected, since n is generally unknown.*

On the basis of the model (22.9), we now introduce the definitions of data-driven SIR and SKR for the feedback control loops given by (22.1)–(22.2).

Definition 22.1 *Given the input–output model (22.9) of the feedback control loop (22.1)–(22.2), the matrices I_s and K_s,*

$$z_s(k) = \begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} = I_s \bar{w}_{2s}(k),\ I_s = \begin{bmatrix} \Gamma_{u,s} L_p & H_{u,\bar{w},s} \\ \Gamma_{y,s} L_p & H_{y,\bar{w},s} \end{bmatrix}, \tag{22.10}$$

$$K_s I_s = 0 \Longrightarrow K_s \begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} = K_s I_s \bar{w}_{2s}(k) = 0,\ K_s \neq 0, \tag{22.11}$$

are called data-driven SIR and SKR of the control loop, respectively.

Note that

$$\begin{bmatrix} \Gamma_{u,s} \\ \Gamma_{y,s} \end{bmatrix} \in \mathcal{R}^{(s+1)(m+p)\times n}, \begin{bmatrix} \Gamma_{u,s} L_p & H_{u,\bar{w},s} \\ \Gamma_{y,s} L_p & H_{y,\bar{w},s} \end{bmatrix} \in \mathcal{R}^{(s+1)(m+p)\times(2s+1)p}.$$

Thus,

$$rank\,(I_s) \leq n + (s+1)p.$$

When $s \geq n$, there exists $K_s \neq 0$, so that (22.11) holds.

The task of our subsequent work is to identify I_s, K_s using sufficient process data, $y(k), u(k), \bar{w}(k)$. Note that $\bar{w}(z) = \hat{V}(z)w(z)$ and $\hat{V}(z)$ is a part of the LCF of the controller. Hence, $\bar{w}(k)$ is known and can be computed online.

22.1.2 Identification of I_s, K_s

Suppose that sufficient process data are collected and ordered into the data sets with the notations introduced in Sub-section 4.4.1,

$$\begin{aligned}
U_f &= U_{k,s} = \begin{bmatrix} u_s(k) & \cdots & u_s(k+N-1) \end{bmatrix} \in \mathcal{R}^{(s+1)p\times N}, \\
Y_f &= Y_{k,s} = \begin{bmatrix} y_s(k) & \cdots & y_s(k+N-1) \end{bmatrix} \in \mathcal{R}^{(s+1)m\times N}, \\
W_f &= W_{k,s} = \begin{bmatrix} \bar{w}_s(k) & \cdots & \bar{w}_s(k+N-1) \end{bmatrix} \in \mathcal{R}^{(s+1)p\times N}, \\
W_p &= W_{k-s-1,s-1} = \begin{bmatrix} \bar{w}_{s-1}(k-s-1) & \cdots & \bar{w}_{s-1}(k-s-2+N) \end{bmatrix} \in \mathcal{R}^{sp\times N}.
\end{aligned}$$

Correspondingly, we have the input–output data set model,

$$\begin{bmatrix} U_f \\ Y_f \end{bmatrix} = \begin{bmatrix} \Gamma_{u,s} \\ \Gamma_{y,s} \end{bmatrix} L_p W_p + \begin{bmatrix} H_{u,\bar{w},s} \\ H_{y,\bar{w},s} \end{bmatrix} W_f + \Psi,$$
$$\Psi = \begin{bmatrix} \alpha_{u,s}(k) & \cdots & \alpha_{u,s}(k+N-1) \\ \alpha_{y,s}(k) & & \alpha_{y,s}(k+N-1) \end{bmatrix} \in \mathcal{R}^{(s+1)(m+p)\times N}.$$

We assume that

- the reference vector $w(k)$ is independent of $r_1(k), r_2(k), r_3(k)$ (for instance, it is a deterministic signal),
- $w(k)$ satisfies the persistently exciting condition

$$rank \begin{bmatrix} W_p \\ W_f \end{bmatrix} = (2s+1)p.$$

On these assumptions, it is clear that

$$\lim_{N\to\infty} \frac{1}{N} \Psi \begin{bmatrix} W_p \\ W_f \end{bmatrix}^T = 0$$

and thus

$$\hat{I}_s = \begin{bmatrix} U_f \\ Y_f \end{bmatrix} \begin{bmatrix} W_p \\ W_f \end{bmatrix}^T \left(\begin{bmatrix} W_p \\ W_f \end{bmatrix} \begin{bmatrix} W_p \\ W_f \end{bmatrix}^T \right)^{-1} \tag{22.12}$$

is an LS and unbiased estimate for I_s.

As introduced in Sect. 4.4, the LS estimate (22.12) can also be computed using the numerically reliable LQ decomposition algorithm, as summarised in the following algorithm.

Algorithm 22.1 *Identification of data-driven SIR in control loops*

Step 0: *Collect data and form W_p, W_f, Y_f, U_f;*
Step 1: *Do an LQ decomposition:*

$$\begin{bmatrix} W_p \\ W_f \\ Z_f \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}, Z_f = \begin{bmatrix} U_f \\ Y_f \end{bmatrix};$$

Step 2: *Set*

$$\hat{I}_s = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1}.$$

It is clear that once $\hat{I}_s$ is found, an estimate $\hat{K}_s$ for K_s can be determined by solving the equation

$$\hat{K}_s \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1} = 0 \Longleftrightarrow \hat{K}_s \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} = 0.$$

And the solution can be parameterised by an SVD of $\begin{bmatrix} L_{31} & L_{32} \end{bmatrix}$,

$$\begin{bmatrix} L_{31} & L_{32} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

with a parameter matrix P as

$$\hat{K}_s = PU_2^T, \, P \neq 0. \tag{22.13}$$

22.2 Recursive SIR and SKR Based Loop Models

In order to develop data-driven algorithms for detecting and recovering performance degradation, we are going to introduce two data-driven models that are derived on the basis of recursive SIR and SKR. These two models are presented in a state space form with accessible state variables.

22.2.1 A Recursive SIR Based Closed-Loop Model

Consider the data-driven SIR model (22.9) and suppose that

$$I_s = \begin{bmatrix} \Gamma_{u,s} L_p & H_{u,\bar{w},s} \\ \Gamma_{y,s} L_p & H_{y,\bar{w},s} \end{bmatrix}$$

has been successfully identified. From the model (22.9), it can be seen that

$$\begin{aligned} y(k-s) &= K_{p,\bar{w}} \, (1:m,:) \, \bar{w}_{s-1}(k-s-1) + D\bar{w}(k-s) + \alpha_y(k-s), \\ K_{p,\bar{w}} &= \Gamma_{y,s} L_p \end{aligned}$$

with $K_{p,\bar{w}} \, (1:m,:)$ denoting the first m rows of $K_{p,\bar{w}}$ and $\alpha_y(k-s)$ being the first m entries of $\alpha_{y,s}(k-s)$. To simplify our study, it is assumed that $D = 0$. We now re-write the above system equation as

$$y(k) = K_{p,\bar{w}} \, (1:m,:) \, \bar{w}_{s-1}(k-1) + \alpha_y(k). \tag{22.14}$$

For the purpose of recovering control performance, an additional dynamic output feedback controller of the following general form is added: $\forall k$,

$$\begin{aligned}\bar{w}(k) &= \sum_{i=1}^{s-1} F_{w,i}\bar{w}\left(k-i\right) + \sum_{j=0}^{s-1} F_{y,i}y\left(k-j\right) \\ &= F_w\bar{w}_{s-2}(k-1) + F_y y_{s-1}(k).\end{aligned} \tag{22.15}$$

Remark 22.3 *It is evident that the above controller is a dynamic system of the* $(s-1)$*-th order. In real applications, the order of the applied controller could be much lower than* $s-1$*. For a controller of the* l*-th order,* $l < s-1$*, the expression of the control law (22.15) can still be adopted with*

$$F_{w,i} = 0,\, F_{y,i} = 0,\, i = l+1, \cdots, s-1.$$

Remark 22.4 *In our subsequent study, notation of the form* $\xi_{s-j}(k-i)$ *is frequently adopted. The reader should be familiar with its definition:*

$$\xi_{s-j}(k-i) = \begin{bmatrix} \xi(k-i-s+j) \\ \vdots \\ \xi(k-i) \end{bmatrix}.$$

The following two recursive equations for $\bar{w}_{s-1}(k)$ with $\bar{w}(k)$ defined in (22.15) are useful in our subsequent work:

$$\begin{aligned}\bar{w}_{s-1}(k) &= \begin{bmatrix} \bar{w}_{s-2}(k-1) \\ \bar{w}(k) \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & F_w & F_y \end{bmatrix} \begin{bmatrix} \bar{w}(k-s) \\ \bar{w}_{s-2}(k-1) \\ y_{s-1}\left(k\right) \end{bmatrix} \\ &=: \begin{bmatrix} F_{w,s-1} & F_{y,s-1} \end{bmatrix} \begin{bmatrix} \bar{w}_{s-1}(k-1) \\ y_{s-1}\left(k\right) \end{bmatrix},\end{aligned} \tag{22.16}$$

$$\begin{aligned}\bar{w}_{s-2}(k) &= \begin{bmatrix} \bar{w}_{s-3}(k-1) \\ \bar{w}(k) \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{w}(k-s+1) \\ \bar{w}_{s-3}(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \bar{w}(k) \\ &=: A_w\bar{w}_{s-2}(k-1) + B_w\bar{w}(k).\end{aligned} \tag{22.17}$$

Let us define $y_{s-1}(k-1)$ as a state variable vector. That is, $y(k-j) \in \mathcal{R}^m$, $j \in [1, s]$, is a sub-vector of the state vector $y_{s-1}(k-1)$,

$$y_{s-1}(k-1) = \begin{bmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k-1) \end{bmatrix} \in \mathcal{R}^{sm}. \tag{22.18}$$

This allows us to write the input–output model (22.14) in the following state space representation form

$$\begin{aligned} y_{s-1}(k+1) &= A_I y_{s-1}(k) + B_I \bar{w}_{s-1}(k) + E_s \alpha_y(k), \qquad (22.19)\\ A_I &= \begin{bmatrix} A_{I,1} \\ 0 \end{bmatrix} \in \mathcal{R}^{sm \times sm},\ A_{I,1} = \begin{bmatrix} 0 & I_{(s-1)m \times (s-1)m} \end{bmatrix},\\ B_I &= \begin{bmatrix} 0 \\ K_{p,\bar{w}}\,(1:m,:) \end{bmatrix} \in \mathcal{R}^{sm \times sp},\ E_s = \begin{bmatrix} 0 \\ I_{m \times m} \end{bmatrix} \in \mathcal{R}^{sm \times m}. \end{aligned}$$

Moreover, together with the dynamics of the controller expressed by (22.17), the overall system dynamics is modelled by

$$\begin{aligned} \begin{bmatrix} y_{s-1}(k+1) \\ \bar{w}_{s-2}(k) \end{bmatrix} &= \begin{bmatrix} A_I & B_{I,1} \\ 0 & A_w \end{bmatrix} \begin{bmatrix} y_{s-1}(k) \\ \bar{w}_{s-2}(k-1) \end{bmatrix} + \begin{bmatrix} B_{I,2} \\ B_w \end{bmatrix} \bar{w}(k) + \begin{bmatrix} E_s \\ 0 \end{bmatrix} \alpha_y(k)\\ &=: A_{yw} \begin{bmatrix} y_{s-1}(k) \\ \bar{w}_{s-2}(k-1) \end{bmatrix} + B_{yw}\bar{w}(k) + \begin{bmatrix} E_s \\ 0 \end{bmatrix} \alpha_y(k), \qquad (22.20)\\ B_I \bar{w}_{s-1}(k) &= \begin{bmatrix} B_{I,1} & B_{I,2} \end{bmatrix} \begin{bmatrix} \bar{w}_{s-2}(k-1) \\ \bar{w}(k) \end{bmatrix}. \end{aligned}$$

The state space model (22.19) will be applied, in the sequel, for performance degradation detection, while (22.20) will be adopted for the online recovery of the performance degradation. The model (22.20) is called (data-driven) recursive SIR of the feedback control loop.

It is of interest to notice that, due to the special forms of A_I, A_w, models (22.19) and (22.20) are stable.

Remark 22.5 *It is worth noting that the dimension of the state vector $y_{s-1}(k-1)$ defined in (22.18) can be selected significantly lower than $s-1$. Let l be some integer smaller than s and satisfy*

$$(l+1)m \geq n.$$

We define

$$y_l(k-1) = \begin{bmatrix} y(k-l-1) \\ \vdots \\ y(k-1) \end{bmatrix} \in \mathcal{R}^{(l+1)m}$$

as the state vector. Correspondingly, the state space model (22.19) becomes

$$\begin{aligned} y_l(k+1) &= A_I y_l(k) + B_I \bar{w}_{s-1}(k) + E_s \alpha_y(k),\\ A_I &= \begin{bmatrix} A_{I,1} \\ 0 \end{bmatrix} \in \mathcal{R}^{(l+1)m \times (l+1)m},\ A_{I,1} = \begin{bmatrix} 0 & I_{lm \times lm} \end{bmatrix},\\ B_I &= \begin{bmatrix} 0 \\ K_{p,\bar{w}}\,(1:m,:) \end{bmatrix} \in \mathcal{R}^{(l+1)m \times sp},\ E_s = \begin{bmatrix} 0 \\ I_{m \times m} \end{bmatrix} \in \mathcal{R}^{(l+1)m \times m}, \end{aligned}$$

and the controller is set to be

$$\bar{w}(k) = \sum_{i=1}^{s-1} F_{w,i}\bar{w}\left(k-i\right) + \sum_{j=0}^{l} F_{y,i}y\left(k-j\right) = F_w\bar{w}_{s-2}(k-1) + F_y y_l(k).$$

As a result, the order of the system eigen-dynamics, $(l+1)m$, becomes (significantly) lower, which is of considerable practical interest for the online implementation. For the sake of simplicity and without loss of generality, in our subsequent study, it is assumed that $l = s-1$.

22.2.2 A Recursive SKR Based Closed-Loop Model

In Sect. 4.4, a data-driven SKR based input–output model has been introduced as follows:

$$\begin{gathered}y_s(k) = K_{p,y}y_{s-1}(k-s-1) + K_{p,u}u_{s-1}(k-s-1) + K_{f,u}u_s(k) + \theta_s(k), \qquad (22.21)\\ \begin{bmatrix} K_p & K_{f,u}\end{bmatrix} = \begin{bmatrix} L_{31} & L_{32}\end{bmatrix}\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22}\end{bmatrix}^{+}, \begin{bmatrix} K_{p,y} & K_{p,u}\end{bmatrix} = K_p,\\ \theta_s(k) \sim \mathcal{N}\left(0, L_{33}L_{33}^T\right).\end{gathered}$$

Analogue to the handling in the previous sub-section, $y(k-s)$ can be written as

$$\begin{aligned}y(k-s) = {} & K_{p,y}\left(1:m,:\right)y_{s-1}(k-s-1) + K_{p,u}\left(1:m,:\right)u_{s-1}(k-s-1)\\ & + K_{f,u}\left(1:m,:\right)u_s(k) + \theta(k-s)\end{aligned}$$

with $K_{p,y}\left(1:m,:\right)$, $K_{p,u}\left(1:m,:\right)$ and $K_{f,u}\left(1:m,:\right)$ denoting the first m rows of $K_{p,y}$, $K_{p,u}$ and $K_{f,u}$, respectively, and $\theta(k-s)$ being the first m entries of $\theta_s(k)$. On the assumption $D = 0$, it holds

$$\begin{aligned}& y(k) = K_{p,y}\left(1:m,:\right)y_{s-1}(k-1) + K_{p,u}\left(1:m,:\right)u_{s-1}(k-1) + \theta(k),\\ & \theta(k) \sim \mathcal{N}\left(0, L_{33}\left(1:m,1:m\right)L_{33}^T\left(1:m,1:m\right)\right).\end{aligned}$$

Let the dynamic output feedback controller of the following general form be applied,

$$\begin{aligned}\forall k, u(k) &= \sum_{i=1}^{s-1} F_{u,i}u\left(k-i\right) + \sum_{j=0}^{s-1} F_{y,i}y\left(k-j\right)\\ &= F_u u_{s-2}(k-1) + F_y y_{s-1}(k). \qquad (22.22)\end{aligned}$$

It yields

$$
\begin{aligned}
u_{s-1}(k) &= \begin{bmatrix} u_{s-2}(k-1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & F_u & F_y \end{bmatrix} \begin{bmatrix} u(k-s) \\ u_{s-2}(k-1) \\ y_{s-1}(k) \end{bmatrix} \\
&=: \begin{bmatrix} F_{u,s-1} & F_{y,s-1} \end{bmatrix} \begin{bmatrix} u_{s-1}(k-1) \\ y_{s-1}(k) \end{bmatrix}
\end{aligned} \tag{22.23}
$$

as well as

$$
\begin{aligned}
u_{s-2}(k) &= \begin{bmatrix} u_{s-3}(k-1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u(k-s+1) \\ u_{s-3}(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u(k) \\
&=: A_u u_{s-2}(k-1) + B_u u(k).
\end{aligned} \tag{22.24}
$$

Remark 22.6 *Analogue to Remark 22.3, we would like to mention that the order of the controller could be set lower than $s-1$.*

By defining $y_{s-1}(k-1)$ as a state variable vector

$$
y_{s-1}(k-1) = \begin{bmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k-1) \end{bmatrix} \in \mathcal{R}^{sm},
$$

we have the following state space representation forms

$$
\begin{aligned}
y_{s-1}(k+1) &= A_K y_{s-1}(k) + B_K u_{s-1}(k) + E_s \theta(k), \qquad (22.25) \\
A_K &= \begin{bmatrix} A_{K,1} \\ K_{p,y}\,(1:m,:) \end{bmatrix} \in \mathcal{R}^{sm \times sm},\ A_{K,1} = \begin{bmatrix} 0 & I_{(s-1)m \times (s-1)m} \end{bmatrix}, \\
B_K &= \begin{bmatrix} 0 \\ K_{p,u}\,(1:m,:) \end{bmatrix} \in \mathcal{R}^{sm \times sp},\ E_s = \begin{bmatrix} 0 \\ I_{m \times m} \end{bmatrix} \in \mathcal{R}^{sm \times m},
\end{aligned}
$$

as well as

$$
\begin{aligned}
\begin{bmatrix} y_{s-1}(k+1) \\ u_{s-2}(k) \end{bmatrix} &= \begin{bmatrix} A_K & B_{K,1} \\ 0 & A_u \end{bmatrix} \begin{bmatrix} y_{s-1}(k) \\ u_{s-2}(k-1) \end{bmatrix} + \begin{bmatrix} B_{K,2} \\ B_u \end{bmatrix} u(k) + \begin{bmatrix} E_s \\ 0 \end{bmatrix} \theta(k) \\
&=: A_{yu} \begin{bmatrix} y_{s-1}(k) \\ u_{s-2}(k-1) \end{bmatrix} + B_{yu} u(k) + \begin{bmatrix} E_s \\ 0 \end{bmatrix} \theta(k), \qquad (22.26) \\
B_K u_{s-1}(k) &= \begin{bmatrix} B_{K,1} & B_{K,2} \end{bmatrix} \begin{bmatrix} u_{s-2}(k-1) \\ u(k) \end{bmatrix}.
\end{aligned}
$$

The model (22.26) is called (data-driven) recursive SKR of the control loop.

At the end of this section, we would like to emphasise that in both models, (22.19) and (22.25), the state vector, $y_{s-1}(k)$, consists of the process output vectors and are thus measured. Moreover, both control inputs, $\bar{w}_{s-1}(k-1)$ and $u_{s-1}(k-1)$, are of the recursive forms given in (22.16) and (22.23), respectively.

22.3 Performance Monitoring and Performance Degradation Recovery

In the subsequent study, we assess the system control performance defined by

$$\begin{aligned} J(i) &= \sum_{k=i}^{\infty} \gamma^{k-i} \left(y^T(k) Q_y y(k) + \bar{w}^T(k) Q_w \bar{w}(k) \right), \\ Q_y &\geq 0, Q_w > 0, 0 < \gamma \leq 1, \end{aligned} \tag{22.27}$$

when (weak) process noises are neglected or by

$$\begin{aligned} J(i) &= \mathcal{E} \sum_{k=i}^{\infty} \gamma^{k-i} \left(y^T(k) Q_y y(k) + \bar{w}^T(k) Q_w \bar{w}(k) \right), \\ Q_y &\geq 0, Q_w > 0, 0 < \gamma < 1, \end{aligned} \tag{22.28}$$

where the influence of process noises on the control performance is taken into account. Since the process models (22.19) ((22.20) as well) and (22.25) have the similar form, we adopt (22.19) as well as (22.20) as the process models under consideration without loss of generality.

In the sequel, we will investigate

- detection of performance degradation, and
- online optimisation of the feedback controller to recover the performance degradation.

22.3.1 *Performance Degradation and Its Detection*

We first consider performance cost function (22.27) and derive a model to describe control performance degradation. To this end, write $J(i)$ as

$$J(i) = y^T(i) Q_y y(i) + \bar{w}^T(i) Q_w \bar{w}(i) + \gamma J(i+1). \tag{22.29}$$

Next, we prove that the solution of (22.29) is given by

$$J(i) = z_{s-1}^T(i) P z_{s-1}(i), \, z_{s-1}(i) = \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-1}(i) \end{bmatrix} \tag{22.30}$$

with P satisfying the following Lyapunov equation

$$P = \gamma \tilde{A}^T P \tilde{A} + R, \tag{22.31}$$

where matrices R and $\tilde{A}$ are given in (22.33) and (22.34), respectively. The proof can be achieved by substituting $J(i)$, $J(i+1)$ given by (22.30) into (22.29), which gives

$$z_{s-1}^T(i)Pz_{s-1}(i) = y^T(i)Q_y y(i) + \bar{w}^T(i)Q_w\bar{w}(i) + \gamma z_{s-1}^T(i+1)Pz_{s-1}(i+1). \tag{22.32}$$

It follows from the control law (22.16) and model (22.19) that

$$\begin{gathered}
\bar{w}(i) = I_w\bar{w}_{s-1}(i),\, y(i) = I_y y_{s-1}(i),\, I_w = \begin{bmatrix} 0 & I_{p\times p}\end{bmatrix},\, I_y = \begin{bmatrix} 0 & I_{m\times m}\end{bmatrix} \\
\Longrightarrow y^T(i)Q_y y(i) + \bar{w}^T(i)Q_w\bar{w}(i) = \begin{bmatrix} y_{s-1}^T(i) & \bar{w}_{s-1}^T(i)\end{bmatrix} R \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-1}(i)\end{bmatrix}, \\
R = \begin{bmatrix} I_y^T Q_y I_y & 0 \\ 0 & I_w^T Q_w I_w \end{bmatrix}, \qquad (22.33) \\
J(i+1) = \begin{bmatrix} y_{s-1}^T(i+1) & \bar{w}_{s-1}^T(i+1)\end{bmatrix} P \begin{bmatrix} y_{s-1}(i+1) \\ \bar{w}_{s-1}(i+1)\end{bmatrix} \\
= \begin{bmatrix} y_{s-1}^T(i) & \bar{w}_{s-1}^T(i)\end{bmatrix} \tilde{A}^T P \tilde{A} \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-1}(i)\end{bmatrix}.
\end{gathered}$$

Here,

$$\tilde{A} = \begin{bmatrix} A_K & B_K \\ F_{y,s-1} & F_{w,s-1}\end{bmatrix}. \tag{22.34}$$

As a result, Lyapunov equation (22.31) holds and $J(i)$ given by (22.30) solves the difference equation (22.29).

For our purpose of performance monitoring, it seems that (22.32) could serve as the performance degradation model. On the other hand, we would like to draw the reader's attention with the following remark.

Remark 22.7 *Equation (22.30) can be understood as the so-called Q-function, which is widely applied in the reinforcement learning technique. It should be, however, remembered that (22.30), as the solution of the cost function (22.27), holds only on the assumption of the control law (22.15) or (22.16). As discussed in Sub-section 20.3.4, the matrix P cannot be well identified using data $\bar{w}_{s-1}(i)$, $y_{s-1}(i)$ and on the basis of (22.32), due to linear relation between $\bar{w}(i)$ and $y_{s-1}(i)$, $\bar{w}_{s-2}(i-1)$ given by (22.15). In fact, the control law (22.15) leads to*

$$\begin{aligned}
J(i) = z_{s-1}^T(i)Pz_{s-1}(i) &= \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-2}(i-1) \\ \bar{w}(i)\end{bmatrix}^T P \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-2}(i-1) \\ \bar{w}(i)\end{bmatrix} \\
&= \begin{bmatrix} y_{s-1}^T(i) & \bar{w}_{s-2}^T(i-1)\end{bmatrix} \begin{bmatrix} I & 0 & (F_y)^T \\ 0 & I & (F_w)^T\end{bmatrix} P \begin{bmatrix} I & 0 \\ 0 & I \\ F_y & F_w\end{bmatrix} \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-2}(i-1)\end{bmatrix}.
\end{aligned}$$

As a result, (22.32) becomes

$$\bar{z}_{s-1}^T(i)\bar{P}\bar{z}_{s-1}(i) = y^T(i)Q_y y(i) + \bar{w}^T(i)Q_w\bar{w}(i) + \gamma\bar{z}_{s-1}^T(i+1)\bar{P}\bar{z}_{s-1}(i+1), \tag{22.35}$$

$$\bar{P} = \begin{bmatrix} I & 0 & \left(F_y\right)^T \\ 0 & I & \left(F_w\right)^T \end{bmatrix} P \begin{bmatrix} I & 0 \\ 0 & I \\ F_y & F_w \end{bmatrix}, \bar{z}_{s-1}(i) = \begin{bmatrix} y_{s-1}(i) \\ \bar{w}_{s-2}(i-1) \end{bmatrix}.$$

We call (22.35) the performance degradation model. In order to identify $\bar{P}$ online, we re-write $J(i)$, analogue to the study in Sect. 20.3.2, into

$$J(i) = \bar{z}_{s-1}^T(i)\bar{P}\bar{z}_{s-1}(i) = \omega^T\phi(i), \tag{22.36}$$

where

$$\omega = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_\eta \end{bmatrix} \in \mathcal{R}^\eta, \eta = (s(p+m)-p+1)(s(p+m)-p)/2,$$

is the parameter vector including all parameters to be identified, and

$$\phi(i) = \begin{bmatrix} \phi_1(i) \\ \vdots \\ \phi_\eta(i) \end{bmatrix} \in \mathcal{R}^\eta, \phi_j(i) \in \mathcal{R}, j = 1, \cdots, \eta,$$

is a vector of time functions consisting of the process data. Note that $\phi_j(i)$ is a scalar function composed of the terms from the set defined below

$$\left\{ \begin{array}{c} y_q(i-\alpha)y_l(i-\beta), q, l = 1, \cdots, m, \alpha, \beta = 0, 1, \cdots, s-1, \\ \bar{w}_q(i-\alpha)\bar{w}_l(i-\beta), q, l = 1, \cdots, p, \alpha, \beta = 1, \cdots, s-1, \\ y_q(i-\alpha)\bar{w}_l(i-\beta), q = 1, \cdots, m, l = 1, \cdots, p, \\ \alpha = 0, 1, \cdots, s-1, \beta = 1, \cdots, s-1, \end{array} \right\}.$$

Here, $y_q(i-\alpha)$, $y_l(i-\beta)$, $\bar{w}_q(i-\alpha)$, $\bar{w}_l(i-\beta)$ are the components of vectors $y(i-\xi)$ and $\bar{w}(i-\xi)$,

$$y(i-\xi) = \begin{bmatrix} \vdots \\ y_\varsigma(i-\xi) \\ \vdots \end{bmatrix} \in \mathcal{R}^m, \bar{w}(i-\xi) = \begin{bmatrix} \vdots \\ \bar{w}_\varsigma(i-\xi) \\ \vdots \end{bmatrix} \in \mathcal{R}^p,$$

$$\xi = \alpha, \beta, \varsigma = q, l.$$

Substituting (22.36) into the performance degradation model (22.35) yields

$$\begin{aligned}&\omega^T\phi(i) = y^T(i)Q_y y(i) + \bar{w}^T(i)Q_w\bar{w}(i) + \gamma\omega^T\phi(i+1) \Longrightarrow\\&\omega^T\left(\phi(i) - \gamma\phi(i+1)\right) = y^T(i)Q_y y(i) + \bar{w}^T(i)Q_w\bar{w}(i).\end{aligned} \tag{22.37}$$

We call (22.37) the regression model of the performance degradation. Based on models (22.35) and (22.37), the following two-step detection algorithm can be performed aiming at detecting performance degradation.

Algorithm 22.2 *Detection of performance degradation*

Step 0: Q_y, Q_w and γ are given, $\bar{P}$ is determined, and sufficient process data are collected;

Step 1: Perform a preliminary performance degradation detection (PPDD) by means of the following detection algorithm:

- *Compute performance residual*

$$\begin{aligned}\Delta = {} & \bar{z}_{s-1}^T(i)\bar{P}\bar{z}_{s-1}(i) - y^T(i)Q_y y(i) - \bar{w}^T(i)Q_w\bar{w}(i)\\&-\gamma\bar{z}_{s-1}^T(i+1)\bar{P}\bar{z}_{s-1}(i+1);\end{aligned} \tag{22.38}$$

- *Run the detection logic*

$$\begin{cases}J_{th,low} \leq \Delta \leq J_{th,high} \Longrightarrow \textit{fault-free} \Longrightarrow \textit{repeat PPDD},\\ \textit{otherwise, faulty} \Longrightarrow \textit{go to the next step},\end{cases}$$

where $J_{th,low}$, $J_{th,high}$ are thresholds;

Step 2: Identify $\bar{P}$ using the regression model (22.37);

Step 3: Detect the performance degradation, based on the identified SPD matrix $\bar{P}$, using a Riemannian distance-based fault detection algorithms presented in Example 3 in Sub-section 15.4.2.

For the use of the above algorithm, we would like to make some noteworthy comments.

The idea behind the two-step detection scheme is to reduce online computations, on the one hand, and to ensure reliable detection on the other hand. In the first step detection, the computation of the performance residual (22.38) is straightforward and less computationally demanding. The threshold setting can be achieved by training using historical process data so that it is robust against the influence of the process and measurement noises. When the condition

$$\Delta < J_{th,low} \text{ or } \Delta > J_{th,high}$$

is satisfied, performance degradation caused by changes in the system dynamics is detected and thus it triggers the second step detection, in which $\bar{P}$ is to be identified. The involved computation in this case is much more demanding than computing Δ, but delivers, on the other hand, rich information about the performance degradation. As discussed in Sect. 15.4, Riemannian distance-based assessment of SPD matrices is a powerful and also helpful tool for us to analyse the change in $\bar{P}$ and, based on it, to make a decision for a right fault-tolerant action.

Next, we study the performance degradation issue with the performance cost function (22.28) and taking into account the process noises. Notice that for $\mathcal{E}\alpha_y(k) = 0$,

$$\begin{aligned} &\mathcal{E}\left(\alpha_y(k)\left[\, y_{s-1}^T(k-1) \;\; \bar{w}_{s-2}^T(k-1)\,\right]\right) \\ &= \mathcal{E}\left(\alpha_y(k)\left[\, y_{s-1}^T(k-1) - \mathcal{E}y_{s-1}^T(k-1) \;\; \bar{w}_{s-2}^T(k-1) - \mathcal{E}\bar{w}_{s-2}^T(k-1)\,\right]\right), \end{aligned} \tag{22.39}$$

and moreover vector

$$\begin{bmatrix} y_{s-1}(k-1) - \mathcal{E}y_{s-1}(k-1) \\ \bar{w}_{s-2}(k-1) - \mathcal{E}\bar{w}_{s-2}(k-1) \end{bmatrix}$$

consists of process and measurement noises. Hence, on the assumption of the involved stochastic process being stationary, the covariance matrix

$$\mathcal{E}\left(\alpha_y(k)\left[\, y_{s-1}^T(k-1) - \mathcal{E}y_{s-1}^T(k-1) \; \bar{w}_{s-2}^T(k-1) - \mathcal{E}\bar{w}_{s-2}^T(k-1)\,\right]\right)$$

is a constant matrix. As a result, it can be proved

$$J(i) = \bar{z}_{s-1}^T(i)\bar{P}\bar{z}_{s-1}(i) + c, \tag{22.40}$$

and furthermore

$$\begin{aligned} \bar{z}_{s-1}^T(i)\bar{P}\bar{z}_{s-1}(i) + c = {} & y^T(i)Q_y y(i) + \bar{w}^T(i)Q_w\bar{w}(i) \\ & + \gamma\left(\bar{z}_{s-1}^T(i+1)\bar{P}\bar{z}_{s-1}(i+1) + c\right). \end{aligned} \tag{22.41}$$

Equation (22.41) is the performance degradation model, based on which performance degradation detection can be achieved. To this end, Algorithm 22.2 can be applied with a slight extension.

22.3.2 Performance Degradation Recovery

As considerable changes in $\bar{P}$ have been detected, which indicate unacceptable trend of system performance degradation, switching an additional controller to the system is an efficient and practical strategy to recover the system performance.

Suppose that the system operation is well described by the model (22.19) or equivalently (22.20). Due to the variation in the system dynamics that leads to the changes in $\bar{P}$, the system matrices become unknown. For our purpose of recovering the system performance, we apply the iterative (updating) Algorithm 20.3 given in Sect. 20.3.4. To this end, we first introduce some useful theoretical results.

For the simplicity of discussion, the cost function (22.27) is considered, which can be further written into

$$J(i)=\sum_{k=i}^{\infty}\gamma^{k-i}\left(y_{s-1}^{T}(k)\bar{Q}_{y}y_{s-1}(k)+\bar{w}^{T}(k)Q_{w}\bar{w}(k)\right),\ \bar{Q}_{y}=I_{y}^{T}Q_{y}I_{y},$$

with I_y defined in (22.27). The optimisation problem is then formulated as

$$\min_{\bar{w}(k)} J(i)$$
$$\text{s.t.}\begin{bmatrix} y_{s-1}(k+1)\\ \bar{w}_{s-2}(k)\end{bmatrix}=A_{yw}\begin{bmatrix} y_{s-1}(k)\\ \bar{w}_{s-2}(k-1)\end{bmatrix}+B_{yw}\bar{w}(k).$$

The optimal solution is given by

$$\bar{w}(k)=\begin{bmatrix} F_y & F_w\end{bmatrix}\begin{bmatrix} y_{s-1}(k)\\ \bar{w}_{s-2}(k-1)\end{bmatrix}=F\bar{z}_{s-1}(k),$$
$$F=-\gamma\left(Q_w+\gamma B_{yw}^{T}P_wB_{yw}\right)^{-1}B_{yw}^{T}P_wA_{yw},\ P_w>0, \tag{22.42}$$

with P_w as the solution of the Riccati equation

$$P_w=A_{yw}^{T}P_wA_{yw}+\tilde{Q}_y-\gamma^{2}A_{yw}^{T}P_wB_{yw}\left(\gamma B_{yw}^{T}P_wB_{yw}+Q_w\right)^{-1}B_{yw}^{T}P_wA_{yw}, \tag{22.43}$$
$$\tilde{Q}_y=\begin{bmatrix}\bar{Q}_y & 0\\ 0 & 0\end{bmatrix}.$$

In order to update the controller online and iteratively to approach the optimal controller (22.42), we can apply the Hewer's algorithm given in Theorem 20.1. As demonstrated in Sect. 20.3.4, using the cost function,

$$J_{j+1}(i)=\sum_{k=i}^{\infty}\gamma^{k-i}\begin{bmatrix} y_{s-1}^{T}(k) & \bar{w}_{s-2}^{T}(k-1)\end{bmatrix}Q_j\begin{bmatrix} y_{s-1}(k)\\ \bar{w}_{s-2}(k-1)\end{bmatrix},$$
$$Q_j=\tilde{Q}_y+F_j^{T}Q_wF_j,$$

which is the performance value corresponding to the controller at the j-th iteration, $F_j\bar{z}_{s-1}$, and the update of the control gain matrix,

$$F_{j+1}=-\gamma\left(Q_w+\gamma B_{yw}^{T}P_{w,j+1}B_{yw}\right)^{-1}B_{yw}^{T}P_{w,j+1}A_{yw},$$
$$P_{w,j+1}=A_{yw,F_j}^{T}P_{w,j+1}A_{yw,F_j}+Q_j,\ A_{yw,F_j}=A_{yw}+B_{yw}F_j, \tag{22.44}$$

we are able to achieve

$$\lim_{j\to\infty} F_j = -\gamma \left(Q_w + \gamma B_{yw}^T P_w B_{yw}\right)^{-1} B_{yw}^T P_w A_{yw}.$$

Recall that the core of Algorithm 20.3 is the identification of $B_{yw}^T P_{w,j} B_{yw}$ and $B_{yw}^T P_{w,j} A_{yw}$ using process data collected online. Below we describe the realisation of the solution for our case. For the sake of simplifying the notation, we drop out the sub-index $j+1$ as well as j.

Suppose that a signal $\vartheta(k)$ is added into the input signal of the existing closed-loop system,

$$\begin{aligned} \bar{w}(k) =& \vartheta(k), \\ \vartheta(k+1) =& \vartheta(k)\rho + \varpi(k), |\rho| << 1, \end{aligned} \tag{22.45}$$

where $\varpi(k)$ is a white noise vector with

$$\mathcal{E}\varpi(k) = 0, \mathcal{E}\varpi(k)\varpi^T(k) = \Sigma_{\varpi},$$

and independent of $y_{s-1}(k)$, $\bar{w}_{s-2}(k-1)$. Consider the cost function

$$\begin{aligned} J(i) &= \mathcal{E}\sum_{k=i}^{\infty} \gamma^{k-i} \left(\bar{z}_{s-1}^T(i)\tilde{Q}\bar{z}_{s-1}(i) + \bar{w}^T(k) Q_w \bar{w}(k)\right), \\ \tilde{Q} &= \tilde{Q}_y + F^T Q_w F, \end{aligned} \tag{22.46}$$

where F is the existing state feedback gain matrix, which, for instance, is equal to F_j after the j-th iteration. Similar to (22.30), we write $J(i)$ as

$$J(i) = \begin{bmatrix} \bar{z}_{s-1}(i) \\ \vartheta(i) \end{bmatrix}^T \tilde{P} \begin{bmatrix} \bar{z}_{s-1}(i) \\ \vartheta(i) \end{bmatrix} + c, \tilde{P} = \begin{bmatrix} \tilde{P}_{11} & \tilde{P}_{12} \\ \tilde{P}_{21} & \tilde{P}_{22} \end{bmatrix}. \tag{22.47}$$

Next, sub-matrices $\tilde{P}_{ij}, i, j = 1, 2$, are determined. Since

$$\begin{aligned} J(i+1) &= \begin{bmatrix} \bar{z}_{s-1}(i+1) \\ \vartheta(i+1) \end{bmatrix}^T \tilde{P} \begin{bmatrix} \bar{z}_{s-1}(i+1) \\ \vartheta(i+1) \end{bmatrix} + c \\ &= \begin{bmatrix} \bar{z}_{s-1}(i) \\ \vartheta(i) \end{bmatrix}^T \begin{bmatrix} A_{yw} & B_{yw} \\ 0 & \rho I \end{bmatrix}^T \tilde{P} \begin{bmatrix} A_{yw} & B_{yw} \\ 0 & \rho I \end{bmatrix} \begin{bmatrix} \bar{z}_{s-1}(i) \\ \vartheta(i) \end{bmatrix} + c, \end{aligned}$$

it turns out

$$\begin{aligned} \tilde{P} &= \begin{bmatrix} \tilde{Q} & 0 \\ 0 & Q_w \end{bmatrix} + \gamma \begin{bmatrix} A_{yw} & B_{yw} \\ 0 & \rho I \end{bmatrix}^T \tilde{P} \begin{bmatrix} A_{yw} & B_{yw} \\ 0 & \rho I \end{bmatrix} \Longrightarrow \\ \tilde{P}_{22} &= Q_w + \gamma \left(B_{yw}^T \tilde{P}_{11} B_{yw} + \rho \tilde{P}_{21} B_{yw} + \rho \left(B_{yw}^T \tilde{P}_{12} + \rho \tilde{P}_{22}\right)\right), \end{aligned}$$

$$\tilde{P}_{21} = \gamma \left(B_{yw}^T \tilde{P}_{11} A_{yw} + \rho \tilde{P}_{21} A_{yw} \right),$$
$$\tilde{P}_{11} = \tilde{Q} + \gamma A_{yw}^T \tilde{P}_{11} A_{yw}.$$

It is evident that for $|\rho| << 1$

$$\tilde{P} = \begin{bmatrix} \tilde{Q} & 0 \\ 0 & Q_w \end{bmatrix} + \gamma \begin{bmatrix} A_{yw} & B_{yw} \\ 0 & \rho I \end{bmatrix}^T \tilde{P} \begin{bmatrix} A_{yw} & B_{yw} \\ 0 & \rho I \end{bmatrix} \Longrightarrow$$
$$\tilde{P}_{22} \approx Q_w + \gamma B_{yw}^T \tilde{P}_{11} B_{yw}, \ \tilde{P}_{21} \approx \gamma B_{yw}^T \tilde{P}_{11} A_{yw},$$
$$\tilde{P}_{11} = \tilde{Q} + \gamma A_{yw}^T \tilde{P}_{11} A_{yw}.$$

That means, identifying $\tilde{P}_{22}, \tilde{P}_{21}$ delivers a good approximation of the (optimal) feedback control gain matrix

$$\begin{aligned} \left[F_y \ F_w \right] : &= -\tilde{P}_{22}^{-1} \left(\tilde{P}_{21} - \left(\tilde{P}_{22} - Q_w F \right) \right) \\ &\approx - \left(B_{yw}^T P_w B_{yw} + \gamma^{-1} Q_w \right)^{-1} B_{yw}^T P_w A_{yw}. \end{aligned} \tag{22.48}$$

Analogue to Algorithm 20.3, we propose the following algorithm for recovering performance degradation.

Remark 22.8 *It is noteworthy to call the reader's attention on the discussion in Sect. 20.3.4, in which it has been illustrated why an additional signal is needed for the identification of matrix $\tilde{P}$.*

Algorithm 22.3 *Data-driven recovery of performance degradation*

Step 0: *Input data: Q_w, $\tilde{Q}$, F_0 (the existing controller to be updated), set $j = 0$ and the tolerance value β;*

Step 1-1: *Set a sufficiently small ρ and generate $\vartheta(k), k = i, \cdots, i + N + 1$, according to (22.45);*

Step 1-2: *Apply the control law*

$$u(k) = F_j \bar{z}_{s-1}(k) + \vartheta(k)$$

and collect process data $\bar{z}_{s-1}(k), k = i, \cdots, i + N + 1$;

Step 1-3: *Identify $\tilde{P}$ using Algorithm 20.2 with data $\bar{z}_{s-1}(k), \vartheta(k), k = i, \cdots, i + N + 1$;*

Step 1-4: *Set $j = j + 1$ and the feedback control gain F_j according to (20.53);*

Step 1-5: *If*

$$\left\| F_j - F_{j-1} \right\|_2 > \beta,$$

go to Step 1-2, otherwise

Step 2: *Output the feedback control gain*

$$F = F_j.$$

22.3.3 Performance Degradation Recovery by Reduced Order Controllers

Recall that the order of the controller (22.15) or (22.22) is (very) high with a lager number $s-1$, which could be too high for a practical implementation. This motivates us to propose an online optimisation algorithm for recovering performance degradation by means of a controller of the lower order. Let

$$0 < l_w < s-1, 0 < l_y < s-1.$$

We consider the following control law:

$$\begin{aligned}\bar{w}(k) &= \sum_{i=1}^{l_w} F_{w,i}\bar{w}\,(k-i) + \sum_{j=0}^{l_y} F_{y,i}\,y\,(k-j) = F_w\bar{w}_{l_w-1}(k-1) + F_y y_{l_y}(k) \\ &= \begin{bmatrix} 0 & F_w \end{bmatrix} \begin{bmatrix} \bar{w}_{s+l_w-2}(k-l_w-1) \\ \bar{w}_{l_w-1}(k-1) \end{bmatrix} + \begin{bmatrix} 0 & F_y \end{bmatrix} \begin{bmatrix} y_{s+l_y-2}(k-l_y-1) \\ y_{l_y}(k) \end{bmatrix} \\ &= \bar{F}_w\bar{w}_{s-2}(k-1) + \bar{F}_y y_{s-1}(k),\ \bar{F}_w = \begin{bmatrix} 0 & F_w \end{bmatrix}, \bar{F}_y = \begin{bmatrix} 0 & F_y \end{bmatrix}. \end{aligned} \tag{22.49}$$

Therefore, our optimisation problem is to find $\bar{F}_w$, $\bar{F}_y$ satisfying the structural restrictions given in (22.49) so that the cost function $J(i)$ given in (22.27) is minimised.

Recall that our performance degradation recovering algorithms have been developed in the well-established framework of the reinforcement learning technique. Our online optimisation strategy can be classified as the so-called policy iteration strategy, which consists of two main steps, (i) policy evaluation, and (ii) policy improvement. To be specific, in Algorithm 22.3, Step 1-1 to Step 1-3 is the realisation of policy evaluation, in which the performance value with respect to the running controller is estimated (predicted), while Step 1-4 results in policy improvement by updating the feedback gain matrix. Notice that due to the structural restriction of $\left(\bar{F}_w, \bar{F}_y\right)$, the feedback gain matrix cannot be determined according to (22.48). In order to solve this problem, we propose the following solution for the realisation of policy improvement.

Denote the controller at the j-th iteration by

$$\bar{w}^j(i) = F_j\bar{z}_{s-1} =: \bar{F}_w^j\bar{w}_{s-2}(i-1) + \bar{F}_y^j y_{s-1}(i).$$

Policy evaluation at the $(j+1)$-th iteration is the value computation of the cost function using the online measurement data collected during operations with controller $\bar{w}^j(i)$. This is achieved by running Step 1-1 to Step 1-3 of Algorithm 22.3. Denote the value of the cost function by $J_{j+1}(i)$, which satisfies

$$\begin{aligned} J_{j+1}(i) &= y^T(i)Q_y y(i) + \left(\bar{w}^j(i)\right)^T Q_w \bar{w}^j(i) + \gamma J_{j+1}(i+1) \\ &= \begin{bmatrix} \bar{z}_{s-1}(i) \\ \bar{w}^j(i) \end{bmatrix}^T \tilde{P}_{j+1} \begin{bmatrix} \bar{z}_{s-1}(i) \\ \bar{w}^j(i) \end{bmatrix}, \end{aligned} \tag{22.50}$$

$$\tilde{P}_{j+1} := \begin{bmatrix} P_{j+1,11} & P_{j+1,12} \\ P_{j+1,21} & P_{j+1,22} \end{bmatrix} \tag{22.51}$$

$$\approx \begin{bmatrix} \gamma A_{yw}^T P_{w,j+1} A_{yw} + \tilde{Q}_y & \gamma A_{yw}^T P_{w,j+1} B_{yw} \\ \gamma B_{yw}^T P_{w,j+1} A_{yw} & \gamma B_{yw}^T P_{w,j+1} B_{yw} + Q_w \end{bmatrix}, \tag{22.52}$$

where $P_{w,j+1}$ is the identified solution of the Lyapunov equation (22.44).

Next, policy improvement at the $(j+1)$-th iteration is to be performed by solving the following optimisation problem:

$$\begin{gathered} \min_{\bar{F}_w^{j+1}, \bar{F}_y^{j+1}} J_{j+1}(i), \\ J_{j+1}(i) = y^T(i)Q_y y(i) + (\bar{w}(i))^T Q_w \bar{w}(i) + \gamma J_{j+1}(i+1), \end{gathered}$$

which yields the update of the controller $\bar{w}^{j+1}(i)$,

$$\begin{gathered} \bar{w}(i) = \bar{F}_w^{j+1} \bar{w}_{s-2}(k-1) + \bar{F}_y^{j+1} y_{s-1}(k) =: \bar{w}^{j+1}(i), \\ \left(\bar{F}_w^{j+1}, \bar{F}_y^{j+1}\right) = \arg \min_{\bar{F}_w^{j+1}, \bar{F}_y^{j+1}} J^{j+1}(i). \end{gathered}$$

Considering the structural restriction of $\left(\bar{F}_w, \bar{F}_y\right)$, we propose to apply the standard gradient descent algorithm to solve the above optimisation problem *iteratively*.

Let

$$F_{j+1} = \begin{bmatrix} F_y^{j+1} & F_w^{j+1} \end{bmatrix}$$

and denote the l-th iteration of F_{j+1} by F_l^{j+1}. It follows from the gradient descent algorithm that

$$F_{l+1}^{j+1} = F_l^{j+1} - \delta \nabla J_{j+1}\left(i, F_l^{j+1}\right), \tag{22.53}$$

where $\nabla J_{j+1}\left(i, F_l^{j+1}\right)$ is the gradient of the cost function $J_{j+1}(i)$ at F_{j+1} and $\delta > 0$ is a tuning parameter. In order to determine $\nabla J_{j+1}\left(i, F_l^{j+1}\right)$, consider (22.50) and write it into

$$\begin{gathered} J_{j+1}(i) = \begin{bmatrix} \bar{z}_{s-1}(i) \\ F\tilde{z}_{l_{yw}}(i) \end{bmatrix}^T \tilde{P}_{j+1} \begin{bmatrix} \bar{z}_{s-1}(i) \\ F\tilde{z}_{l_{yw}}(i) \end{bmatrix}, \\ F = \begin{bmatrix} F_y & F_w \end{bmatrix}, \tilde{z}_{l_{yw}}(i) = \begin{bmatrix} y_{l_y}(i) \\ \bar{w}_{l_w-1}(i-1) \end{bmatrix}. \end{gathered}$$

It turns out

$$
\begin{aligned}
\frac{\partial J_{j+1}(i)}{\partial F} &= 2\frac{\partial \tilde{z}_{l_{yw}}^T(i)F^T P_{j+1,21}\bar{z}_{s-1}(i)}{\partial F} + \frac{\partial \tilde{z}_{l_{yw}}^T(i)F^T P_{j+1,22}F\tilde{z}_{l_{yw}}(i)}{\partial F} \\
&= 2\frac{\partial \bar{z}_{s-1}^T(i)\bar{I}^T F^T P_{j+1,21}\bar{z}_{s-1}(i)}{\partial F} + \frac{\partial \bar{z}_{s-1}^T(i)\bar{I}^T F^T P_{j+1,22}F\bar{I}\bar{z}_{s-1}(i)}{\partial F}, \\
\bar{I} &= \begin{bmatrix} 0 & I \end{bmatrix}.
\end{aligned}
$$

According to the rules

$$
\begin{aligned}
\frac{\partial tr\,(AB)}{\partial A} &= B^T, \\
\frac{\partial tr\left(ABA^T C\right)}{\partial A} &= CAB + C^T AB^T,
\end{aligned}
$$

it holds

$$
\begin{aligned}
\frac{\partial \bar{z}_{s-1}^T(i)\bar{I}^T F^T P_{j+1,21}\bar{z}_{s-1}(i)}{\partial F} &= P_{j+1,21}\bar{z}_{s-1}(i)\bar{z}_{s-1}^T(i)\bar{I}^T, \\
\frac{\partial \bar{z}_{s-1}^T(i)\bar{I}^T F^T P_{j+1,22}F\bar{I}\bar{z}_{s-1}(i)}{\partial F} &= 2P_{j+1,22}F\bar{I}\bar{z}_{s-1}(i)\bar{z}_{s-1}^T(i)\bar{I}^T.
\end{aligned}
$$

As a result, the iteration computation of (22.53) is

$$
F_{l+1}^{j+1} = F_l^{j+1} - 2\delta\left(P_{j+1,21} + P_{j+1,22}F_l^{j+1}\bar{I}\right)\bar{z}_{s-1}(i)\bar{z}_{s-1}^T(i)\bar{I}^T. \tag{22.54}
$$

Using the Kronecker product and the associated operation rules, the vectorised form of (22.54) is given by

$$
\begin{aligned}
vec\left(F_{l+1}^{j+1}\right) &= vec\left(F_l^{j+1}\right) - 2\delta\left(Z(i)\otimes P_{j+1,22}\right)vec\left(F_l^{j+1}\right) - 2\delta vec\left(\Psi(i)\right) \\
&= \left(I - 2\delta\left(Z(i)\otimes P_{j+1,22}\right)\right)vec\left(F_l^{j+1}\right) - 2\delta vec\left(\Psi(i)\right), \\
Z(i) &= \bar{I}\bar{z}_{s-1}(i)\bar{z}_{s-1}^T(i)\bar{I}^T, \Psi(i) = P_{j+1,21}\bar{z}_{s-1}(i)\bar{z}_{s-1}^T(i)\bar{I}^T.
\end{aligned}
$$

Although $P_{j+1,22}$ is in general positive definite, matrix $Z(i)$ is positive semi-definite. Consequently, matrix $Z(i)\otimes P_{j+1,22}$ is also positive semi-definite, a well-known result from the framework of the Kronecker product. In other words, some of the eigenvalues of matrix

$$
I - 2\delta\left(Z(i)\otimes P_{j+1,22}\right)
$$

are equal to one for all possible $\delta > 0$. In order to guarantee the iteration convergence, we suggest the following simple solution. At first, sufficient data are collected, $\bar{z}_{s-1}(i), \cdots, \bar{z}_{s-1}(i+N)$. It is straightforward that the optimisation problem

$$\begin{aligned}&\min_F \frac{1}{N+1}\sum_{k=0}^{N} J_{j+1}(i+k)\\ &= \min_F \frac{\sum\limits_{k=0}^{N}\begin{bmatrix}\bar{z}_{s-1}(i+k)\\ F\tilde{z}_l(i+k)\end{bmatrix}^T P_{w,j+1}\begin{bmatrix}\bar{z}_{s-1}(i+k)\\ F\tilde{z}_l(i+k)\end{bmatrix}}{N+1}\end{aligned}$$

can be solved by the following iteration algorithm:

$$F_{l+1}^{j+1} = F_l^{j+1} - 2\delta\left(P_{j+1,21} + P_{j+1,22}F_l^{j+1}\bar{I}\right)\frac{\sum\limits_{k=0}^{N}\left(\bar{z}_{s-1}(i+k)\bar{z}_{s-1}^T(i+k)\bar{I}^T\right)}{N+1}. \tag{22.55}$$

On the assumption that the process data are sufficiently excited so that the matrix

$$\bar{I}\sum_{k=0}^{N}\left(\bar{z}_{s-1}(i+k)\bar{z}_{s-1}^T(i+k)\right)\bar{I}^T$$

is positive definite, the matrix

$$\bar{Z}(i)\otimes P_{j+1,22},\, \bar{Z}(i) = \frac{1}{N+1}\bar{I}\sum_{k=0}^{N}\left(\bar{z}_{s-1}(i+k)\bar{z}_{s-1}^T(i+k)\right)\bar{I}^T,$$

is positive definite. This guarantees the iteration

$$\begin{aligned}vec\left(F_{l+1}^{j+1}\right) &= vec\left(F_l^{j+1}\right) - 2\delta\left(\bar{Z}(i)\otimes P_{j+1,22}\right)vec\left(F_l^{j+1}\right) - 2\delta vec\left(\bar{\Psi}(i)\right)\\ &= \left(I - 2\delta\left(\bar{Z}(i)\otimes P_{j+1,22}\right)\right)vec\left(F_l^{j+1}\right) - 2\delta vec\left(\bar{\Psi}(i)\right),\\ \bar{\Psi}(i) &= P_{j+1,21}\sum_{k=0}^{N}\left(\bar{z}_{s-1}(i+k)\bar{z}_{s-1}^T(i+k)\right)\bar{I}^T,\end{aligned}$$

converges by selecting δ satisfying

$$0 < \delta < \frac{1}{\lambda_{\max}\left(\bar{Z}(i)\otimes P_{j+1,22}\right)}.$$

Here, $\lambda_{\max}\left(\bar{Z}(i)\otimes P_{j+1,22}\right)$ is the maximum eigenvalue of $\bar{Z}(i)\otimes P_{j+1,22}$.

Remark 22.9 *The value,*

$$\frac{1}{N+1}\sum_{k=0}^{N} J_{j+1}(i+k),$$

can be interpreted as an approximation of $\mathcal{E} J_{j+1}(i)$*. Thus, the modified solution is of certain robustness against noises.*

Readers may notice that the online computation of $\lambda_{\max}\left(\bar{Z}(i) \otimes P_{j+1,22}\right)$ would be challenging due to the possibly high dimension of matrix $\bar{Z}(i) \otimes P_{j+1,22}$. One possibility is to set δ sufficient small. On the other hand, this will lead to a considerably low convergence rate. We would like to draw the reader's attention to the discussion in Sect. 14.6.2. It has been demonstrated that for a positive semi-definite matrix $A \in \mathcal{R}^{m \times m}$, it holds

$$\begin{aligned} \lambda_{\max}(A) &\leq \|A\|_{\infty} = \|A\|_1, \\ \|A\|_{\infty} &= \max_{1 \leq l \leq m} \sum_{j=1}^{m} \left|a_{lj}\right|, A = \left(a_{lj}\right), l, j = 1, \cdots, m. \end{aligned} \tag{22.56}$$

Because the computation for $\|A\|_{\infty}$ can be well online performed, as given in (22.56), we suggest to set δ as

$$0 < \delta < \frac{1}{\left\|\bar{Z}(i) \otimes P_{j+1,22}\right\|_{\infty}}. \tag{22.57}$$

We now summarise the major results on recovering performance degradation using a reduced order controller as an algorithm.

Algorithm 22.4 *Data-driven recovery of performance degradation by a reduced order controller*

Step 0: *Set* $j = 0$, *and collect process data;*
Step 1: *Identify* $\tilde{P}$ *using Algorithm 22.3 (Step 1-1 to Step 1-3) and set*

$$P_{w,j+1} = \tilde{P};$$

Step 2: *Collect data and build* $\bar{Z}(i) \otimes P_{j+1,22}$, *determine* $\left\|\bar{Z}(i) \otimes P_{j+1,22}\right\|_{\infty}$ *according to (22.56) and further set* δ *according to (22.57);*
Step 3: *Run iterative algorithm (22.55) for* F_{j+1}, *and check the iteration convergence. If not, go to the next step, otherwise stop;*
Step 4: *Set* $j = j + 1$ *and go to Step 1.*

It is remarkable that Algorithm 22.4 enables online performance recovery using any LTI dynamic output feedback controller.

22.4 Notes and References

Although the main objective of this chapter is to study fault-tolerant control issues in the data-driven fashion, we have begun with closed-loop identification of data-driven SIR and SKR. The background of this work is the fact that fault-tolerant control becomes urgently necessary if the existing controller cannot deliver satisfactory control performance. In this case, when an identification of the process model is performed with the existing controller, this has to be realised in the closed-loop configuration. A further aspect is that the so-called data-driven forms of SIR and SKR can be directly identified using the process data collected online and, based on them, a data-driven re-configuration of an (optimal) controller can be handled in a systematic manner, see also the discussion in Sect. 21.2.

System identification in feedback control loops is a classic topic in control theory and engineering [1, 2]. A challenging issue in this thematic field is the handling of correlations between the measurement noise and the process control variables due to the feedback effect. In the (very) few publications related to the identification of data-driven SIR and SKR in closed-loops [3, 4], this issue has not been systematically addressed. Consequently, the estimated data-driven SIR and SKR may not be free of bias. On the basis of the observer-based input–output model introduced in Sect. 19.1, we have proposed in the first section of this chapter an algorithm that allows us to achieve a bias-free identification of the data-driven form of SIR and SKR in a closed-loop.

The state space models of the data-driven SIR, (22.19) and (22.20), and SKR, (22.25) as well as (22.26), have been derived based on the recursive SIR and SKR. These models can be either identified and then adopted for the controller and observer design or serve as a basis for the online performance monitoring and performance degradation recovery. The most convincing argument for applying these models for performing online monitoring and control tasks is that all state variables are accessible. They are indeed the process output variables in the models (22.19) and (22.25), and output as well as input variables in the models (22.20) and (22.26). This allows us to apply the existing algorithms presented in Chaps. 20–21 to achieving performance monitoring and performance degradation recovering. On the other hand, it should be noticed that the dimension of these state space models is considerably high. Also, the structural properties like controllability and observability have not been investigated. In fact, our intention for this study is to build a framework, in which further investigations on, for instance, model reduction, structural analysis etc. can be well carried out.

Based on the above state space models, performance degradation detection and recovery algorithms have been developed. They are the immediate application of the algorithms proposed in Chaps. 20–21 with some straightforward extensions. It should be noticed that this is only possible when the controller is of the form (22.15) or (22.22) with a (very) higher system order. In order to achieve performance degradation recovery by means of a reduced controller, we have proposed, in the last part of our work, an iteration algorithm for the policy adaptation. From the optimisation point of view, we have applied the standard gradient descent method [5] for the controller optimisation with constraints on the structure of the feedback gain matrix, induced by the restriction on the order of the controller. Based on a convergence analysis of the applied gradient descent algorithm, we have further proposed a modified version for the iterative optimisation with a guarantee of the iteration convergence. In this work, some well-known rules for the derivative computations of matrix trace

[6] and Kronecker product as well as some associated computations [7, 8] have been applied.

We would like to mention that our work in Sect. 22.3.2 on the performance degradation recovery can be viewed as an alternative realisation of the Q-learning aided LQR controller optimisation using output data (instead of the direct measurement of the state variables). In the literature [9, 10], the popular strategy to deal with this issue has been reported, which generally consists of a two-step procedure:

- approximation of the state variables by means of the system input and output data, and based on it,
- the optimisation problem is solved in a similar way like the original LQR optimisation algorithm.

As we know from the parity-space approach and its state space realisation in form of an observer [11], the first step is in fact a dead-beat observer. Consequently, we can understand this optimisation strategy as a deadbeat observer-based LQ control, whose performance, in comparison with the $\mathcal{H}_2$ control scheme, is obviously less than optimal. In our alternative scheme proposed in Sect. 22.3.2, we have handled this problem in a different way. The key step is the (data-driven) model (22.19), which enables an optimisation of a dynamic output controller with flexible structure.

References

1. B. Huang and R. Kadali, *Dynamic Modelling, Predictive Control and Performance Monitoring, a Data-Driven Subspace Approach*. London: Springer-Verlag, 2008.
2. T. Katayama, *Subspace Methods for System Identification*. London: Springer-Verlag, 2005.
3. K. Koenings, M. Krueger, H. Lou, and S. X. Ding, “A data-driven computation method for the gap metric and the optimal stability margin,” *IEEE Trans. on Automatic Control*, vol. 63, pp. 805–810, 2018.
4. H. Luo, S. Yin, T. Liu, and A. Khan, “A data-driven realization of the control-performance-oriented process monitoring system,” *IEEE Trans. on Industrial Electronics*, vol. 67, pp. 521–530, 2020.
5. D. Bertsekas, *Nonlinear Programming, 2nd Edition*. Athena Scientific, 1999.
6. K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. internet version November, 2008: http://matrixcookbook.com, 2008.
7. J. Brewer, “Kronecker products and matrix calculus in system theory,” *IEEE Trans. on Circuits and Systems*, vol. 25, pp. 772–781, 1978.
8. A. Graham, *Kronecker Products and Matrix Calculus with Applications*. Chichester: Ellis Horwood Limited, 1981.
9. F. L. Lewis and K. G. Vamvoudakis, “Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data,” *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 41, pp. 14–25, 2011.
10. C. Hua, *Reinforcement Learning Aided Performance Optimization of Feedback Control Systems*. PhD dissertation, University of Duisburg-Essen, 2020.
11. S. X. Ding, *Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools, 2nd Edition*. London: Springer-Verlag, 2013.

Index

S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*,
https://doi.org/10.1007/978-3-662-62004-5