Distributed Fault Detection for Large-Scale and Interconnected Systems

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

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

genehmigte Dissertation

von

Jiarui Zhang
aus
Shandong, V.R. China

Gutachter: Prof. Steven X. Ding
Gutachter: Prof. Linlin Li
Tag der mündlichen Prüfung: 01.08.2023

Preface

This work has been done at the Institute for Automatic Control and Complex Systems (AKS) in the Faculty of Engineering at the University of Duisburg-Essen, Germany. First of all, I am honored to express my sincere gratitude to my supervisor, Prof. Dr.-Ing. S.X. Ding, for his exceptional guidance and support throughout my scientific research work. I would like to extend my sincerest thanks to my supervisor for his unwavering dedication and understanding during my Ph.D. studies. His well-informed direction and valuable research insights have been helpful in guiding me to produce my work. And I am particularly appreciative of his sharing of his vast academic and scientific writing experience, which is important in assisting me to complete my dissertation with confidence. My sincere appreciation must also go to Prof. Changbin Hu for his insightful guidance and productive discussions regarding this thesis. I would also like to thank Prof. Dr.-Ing. Linlin Li and Dr. Peng Xin for their valuable support and input on matters related to this work. I am truly appreciative of their continuous encouragement and support.

I would like to thank my colleagues at AKS, Dr.-Ing. Birgit Kppen-Seliger, Dr.-Ing. Chris Louen, Dr.-Ing. Lu Qian, Dr.-Ing. Changsheng Hua, Dr.-Ing. Ting Xue, Dr.-Ing. Yuhong Na, M.Sc. Tianyu Liu, M.Sc. Yannian Liu, M.Sc. Micha Obergfell, M.Sc. Hogir Rafiq, M.Sc. Abdul Salah, M.Sc. Tieqiang Wang, M.Sc. Deyu Zhang, M.Sc. Caroline Charlotte Zhu, M.Sc. Liutao Zhou, M.Sc. Wei Cheng and M.Sc. Ketian Liang for the wonderful cooperation, valuable discussion, and helpful suggestions. Your support and contributions have been invaluable to this work and I am grateful for the opportunity to have worked with you. In addition, special thanks to Prof. Lijia Luo, Dr. Dong Zhao, Dr. Junwei Zhu, Dr. Yang Zhang, Dr. Xiaolu Chen, Dr. Nana Zhou, Dr. Guangjie Chen, Dr. Yun Feng, Dr. Bin Guo, Dr. Wei He, Dr. Han Yu, Dr. Yan Li, Dr. Yang Song, Dr. Xiuli Wang, Dr. Yang Chao, Dr. Ruijie Liu, Ms. He Li who offered their valuable suggestions and support to my research topic during their visits in AKS. My acknowledgement will be incomplete without thanking Mrs. Sabine Bay, Dipl.-Ing. Klaus Gbel, Mr. Ulrich Janzen and M.Sc. Michael Baumann for the help and support regarding our organizational responsibilities. Your dedication and commitment have been invaluable in ensuring the smooth running of our operations.

Finally, I would like to extend my sincerest and heartfelt gratitude to my beloved family for their unwavering love and support. They have been a constant source of strength and encouragement throughout my life. Thank you from the bottom of my heart.

Aachen, 08.11.2023 Jiarui Zhang

To my parents

Contents

List of Figures

List of Tables

List of Notations

Abbreviations

Abbreviation	Expansion
AKS	Automatic Control and Complex Systems
FD	fault detection
DO	diagnostic observer
PSA	parity space approaches
FDF	fault detection filter
GLR	generalized likelihood ratio
PCA	principal component analysis
PLS	partial least squares
CCA	canonical correlation analysis
SVM	support vector machine
NN	neural network
MVA	multivariate analysis
LMI	linear matrix inequality
LS	least square
FAR	false alarm rate
MDR	missed detection rate
FDR	fault detection rate
LTI	linear time invariant
DCF	doubly co-prime factorization
RCF	right co-prime factorization
LCF	left co-prime factorization
RMS	root mean square
SIM	subspace identification method
SVD	singular value decomposition
DARE	discrete time algebraic Riccati equation
MTTD	mean time to detection

Mathematical notations

Notation	Description
$\forall$	for all
$\sim$	is distributed as
$\in$	belong to
$\notin$	does not belong to
$\mathcal{R}^n$	space of real n-dimensional vectors
$\mathcal{R}^{n\times m}$	space of real n by m matrices
$\Pr(A)$	the probability of random event A
$\mathcal{N}(a, \Sigma)$	Gaussian distribution with mean vector a and covariance matrix Σ
$\chi^2(m)$	chi-squared distribution with m degrees of freedom
$\chi^2_\alpha(m)$	inverse cumulative distribution function of the chi-squared distribution with m degrees of freedom and probability value α
$\mathrm{E}(x)$	expected value of random variable x
$\mathrm{var}(x)$	variance of random variable x
$\mathrm{cov}(x, y)$	covariance matrix between random variables x and y
$X > 0$	X is positive definite matrix
$X \geqslant 0$	X is positive semi-definite matrix
X^T	transport of X
X^{-1}	inverse of X
X^-	left or right inverse of matrix X
$\mathrm{rank}(X)$	rank of matrix X
$\sigma_i(X)$	the i-th singular value of matrix X
$\rho(A)$	spectral radius of matrix A
$\lambda_{max}(A)$	maximal eigenvalue of matrix A
$\mathrm{tr}(A)$	trace of matrix A
$\langle a, b\rangle$	inner product with $\langle a, b\rangle = a^\top b$
$\langle a, b\rangle_W$	weighted inner product with $\langle a, b\rangle_W = a^\top W b$
$\mathcal{RH}_\infty$	the set of all proper and real rational stable transfer function matrices
δ_{ij}	delta function with $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
$\mathrm{card}(\mathcal{N})$	cardinality of set $\mathcal{N}$
vector norm	
$\vert\vert \cdot \vert\vert$	Euclidean norm of a vector
$\vert\vert \cdot \vert\vert_W$	weighted Euclidean norm of a vector with $\vert\vert a\vert\vert_W = \sqrt{a^T W a}$
matrix norm	

$\|\| \cdot \|\|_1$	$1-$norm of a matrix
$\|\| \cdot \|\|_2$	$2-$norm of a matrix
$\|\| \cdot \|\|_\infty$	$\infty-$norm of a matrix
signal norm	
$\|\| \cdot \|\|_2$	$\mathcal{L}_2$ norm of a signal
system norm	
$\|\|G\|\|_2$	$\mathcal{H}_2$ norm of stable transfer function matrix G
$\|\|G\|\|_\infty$	$\mathcal{H}_\infty$ norm of stable transfer function matrix G

1 Introduction

For the purpose of maximizing customer satisfaction, performing more flexible and sophisticated tasks, and increasing the quality of industrial products in modern industrial practices, large-scale and complex systems are becoming a pervasive component of modern industrial processes over the past decades. Such systems can

- have digital operating units to gain robustness against external disturbance,
- be distributed over space,
- be decomposed into a series of subsystems, which are connected by physical links or communication channels.

Given the aforementioned characteristics of large-scale systems, ensuring safety and reliability is of paramount importance, as faults or failures in a single subsystem can have a cascading effect, leading to potentially catastrophic outcomes for the entire system. For example,

- Air France Flight 447 was a scheduled international passenger flight from Rio de Janeiro, Brazil, to Paris, France. On 1 June 2009, the aircraft suffered temporary inconsistencies between the airspeed measurements, which is likely influenced by ice crystals obstructing the aircraft's pilot tubes and caused the autopilot to disconnect. After the inconsistencies, the crew reacted incorrectly and ultimately caused the aircraft to enter an aerodynamic stall. Eventually, the aircraft crashed into the Atlantic Ocean, which killed all 228 passengers and crew on board [38].
- On April 20, 2010, due to the defective cement on the well, high-pressure methane gas from the well expanded into the marine riser and rose into the drilling rig, where it ignited and exploded, engulfing the Deepwater Horizon oil platform. The accident killed 11 workers, injured 17, released about 4.9 million barrels (210 million US gal; 780,000 m^3) of crude oil into the ocean and is considered as the largest marine oil spill. Both the spill and the clean-up efforts had effects on the environment [128].

Other examples can be found in [51, 100]. From the above examples, the traditional way of enhancing the quality, reliability, and robustness of individual system components like sensors, actuators, controllers, and computers can't guarantee that no fault would occur during the entire lifetime operation of the processes. And if the fault is not detected and handled promptly in time, it may not only affect the system operation but also result in significant consequences (e.g., human safety, major economic effects, and environmental impact). Motivated by these observations, automatized fault detection (FD) techniques are becoming one of the indispensable parts of large-scale and complex systems and is the essential step to ensuring reliable treatment of undesirable events. Over the past decades, there is a growing body of literature that recognises the importance of FD techniques both in the academic and engineering domains and the topics of FD are becoming an active research area in the control community. This chapter reviews some of the major developments and progresses in FD and then introduces the motivations and objectives of this thesis.

1.1 Basic Concepts of Fault Detection

Consider a dynamic system as sketched in Figure 1.1. It consists of actuator, process, and sensor. For given control signals, the actuator converts them into corresponding operations

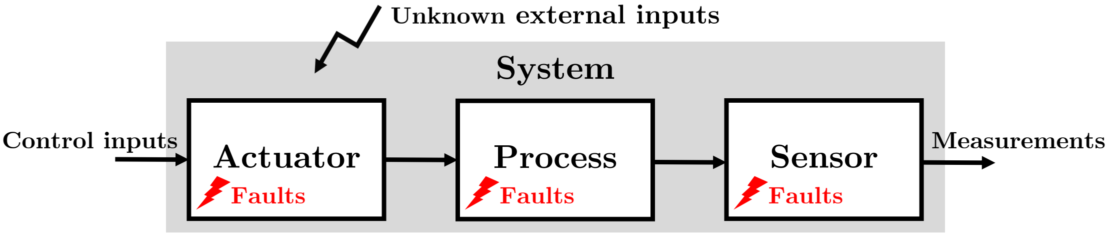


Figure 1.1: System configuration

such that the process can be controlled into desired operating conditions. The sensor is the element to capture data, which provides information on the running conditions of the process [41]. In real applications, all these elements are usually influenced by some unknown external inputs, which can be random noises and/or deterministic disturbances caused by the running environment around the system. Generally, a fault is an unpermitted deviation of at least one characteristic property (feature) of the system from acceptable, usual, and standard condition [59]. Faults can occur in almost all systems [52] and also in each functional element. The essential task of an FD method is to detect the occurrence of faults, which may lead to undesired or intolerable behaviour of the whole system, in the functional elements of the process in time and to give a quick alarm under the influence of unknown external inputs [24].

Motivated by the requirements of the safety, reliability, and economic efficiency on modern industrial processes, over the past decade, there has been a significant increase in the development of FD techniques across various fields. Also, this development has become a crucial aspect of modern industrial processes and has been actively researched in both academic and industrial areas. Most of the existing FD methods follow the centralized strategy, where all measurements are collected in a central station to perform the corresponding FD actions and can be roughly classified into categories [24, 25, 43, 44]

- **FD based on signal processing**: If specific process signals contain rich information about the faults of interest, an appropriate signal processing can convey this information in the form of characteristics or symptoms to monitor the process [70, 80]. Time domain symptoms include things like magnitudes, limits, trends, statistical moments of the amplitude distribution or envelope, and frequency domain symptoms include things like spectrum power densities, frequency spectral lines, spectrum, etc. The fundamental concept behind this kind of technique is to generate symptoms from sensor measurements, separate fault information from symptoms and make a judgement about a fault based on that. Traditional methods based on time- and frequency-domain analysis are primarily used, including synchronous averaging [50, 81, 124] and Fourier analysis [77, 82, 88]. Since the resolutions of time- and frequency-domain need to be balanced under certain conditions, wavelet analysis [11, 81] and empirical mode decomposition [69, 107, 108] are two recent studies that concentrate on time-frequency analysis. The traditional signal processing-based techniques are primarily applicable to linear stationary processes. For dynamic systems, signal-processing-based approaches are mostly employed for those operations in the steady state [24].

- **FD based on hardware redundancy**: Technically speaking, the so-called redundancy is crucial to the successful detection of faults. One of the most common methods of creating system redundancy is through hardware redundancy as shown in Figure 1.2. It rebuilds the essential components of the monitored system by direct hardware duplication with the goal of enhancing system reliability. If the output of the process component differs from one of its identical redundancy, faults in the process component are then discovered. In some circumstances, the defect can also be immediately isolated. High reliability and direct fault isolation are the main appealing features of this technique. However, the implementation of hardware redundancy is only allowed in the case of a few important components in safety-critical industries (e.g., nuclear, aviation, and aerospace systems [36, 85, 97]) due to their greater reconstruction costs.

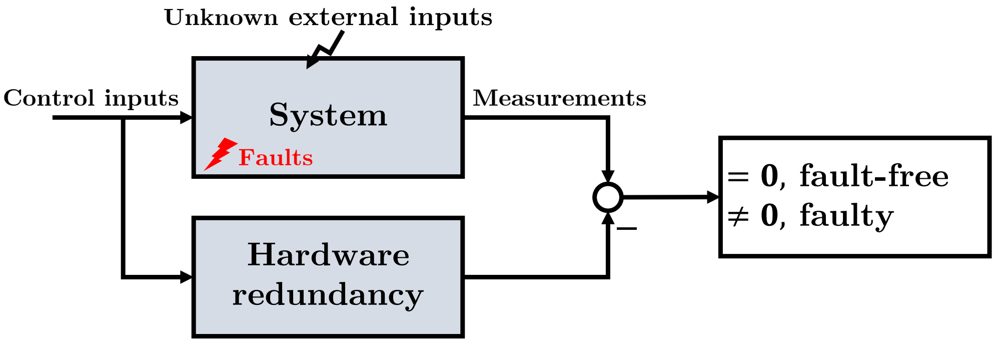


Figure 1.2: Fault detection based on hardware redundancy

- **Model-based FD**: The basic concept behind model-based FD approaches is to substitute a process model, which is based on prior knowledge about the mathematical input-output relation or state space model of the systems and is implemented in form of software, for expensive hardware redundancy [16, 83]. Similar to hardware redundancy methods, the process model runs concurrently with the process and is stimulated by the same process inputs. It is reasonable to anticipate that the outputs of the model will closely match the measurements from the real process in a fault-free operating state and will clearly deviate in the presence of a process fault. Hence the residual generator is built to obtain the so-called residual, which is the discrepancy between the measured real process variables and their estimates from the model, for detecting fault as shown in Figure 1.3. The residual signal conveys the essential information for faults and if the model is perfect, faults will be alarmed when the residual signal doesn't equal zero. However, it is almost impossible to build a perfect model and thus the residual signal is often influenced by model uncertainties and unknown inputs. Therefore, additional residual evaluation is required to extract the information about fault from distorted residuals. Beard and Jones started research on model-based methodologies in the early 1970s. The diagnostic observer (DO) [33, 72, 74, 87, 111], parity space approaches (PSA) [35, 45, 46, 99] and fault detection filter (FDF) [34, 63, 65, 91] are some of the popular model-based techniques used today. Numerous sophisticated FD techniques are created on the basis of these methodologies to address problems with robustness against unknown input and model uncertainties, and optimal designs. Reviews and analyses of the basic and advanced model-based fault diagnosis methodologies under development are provided in [2, 31, 58, 105, 125].

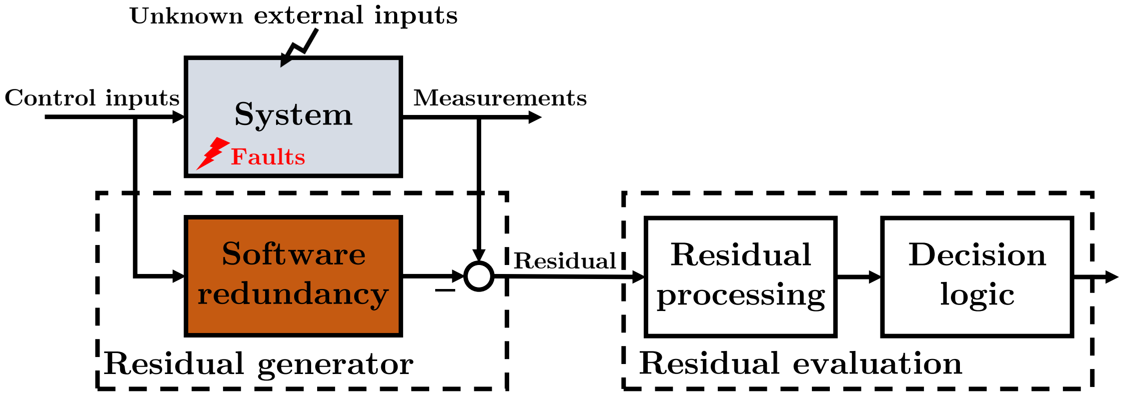


Figure 1.3: Model-based fault detection

- **Data-driven FD**: Building the aforementioned process model is frequently impractical, time-consuming, and expensive in engineering for many large-scale industrial applications, particularly in chemical processes, power grids, and high-speed trains. This restricts the use of model-based FD techniques. Inspired by this trend, researchers have shown an increased interest in data-driven strategies, which use a substantial amount of process data as a priori knowledge instead of system models, to build FD approaches. These approaches first extract an FD system from historical data sets in the training phase. Then the trained FD system processes online measurement data to detect fault. Similar to methods based on signal processing, this technique is mostly applied in static processes [25]. Since most of the data-driven approaches assume that the process data have certain statistical properties, generalized likelihood ratio (GLR) test and Neyman-Pearson Lemma are applied to handle the issue for optimal data-driven FD [27, 84]. In addition, those well-known multivariate analysis methods like principal component analysis (PCA) [3, 15, 37, 115, 118] , partial least squares (PLS) [40, 47, 55, 78, 79] and canonical correlation analysis (CCA) [14, 18, 20, 120] have been proposed to perform FD in certain situations. Notice that PCA and CCA are special cases of the GLR-based solution and PLS is a recursive realization of CCA [25, 27]. With strong capabilities in function approximation and adaptive learning, some machine learning methods play an important role in nonstatistical data-driven FD, e.g., support vector machine (SVM) [96, 109, 116, 117] and neural network (NN) [5, 101, 103].

- **Knowledge-based FD**: The knowledge-based methods are also known as qualitative model-based methods, they utilize the qualitative models as a priori to build FD systems [104]. Those approaches performed based on the evaluation of online monitored data in terms of a set of logic rules, which is learned effectively by human experts from their knowledge and experience, to mimic the cognitive behaviour of human experts [1, 43, 102]. Compared with the above methods, knowledge-based methods have advantages such as ease of development, transparent reasoning, and the capability to explain the solutions. But they are system specific and thus have low generality and low expandability [43]. Nowadays, the so-called task-based diagnosis expert system has been proposed to provide a general and flexible solution for FD systems [8, 67].

For large-scale systems, it is hard to use one sensor to monitor the entire process and thus they are usually equipped with a bunch of sensors. Traditionally, each sensor monitors only local and limited process information and then all measured process variables (process data) are collected in a central station to perform the corresponding FD actions. Such centralized handling and performing of FD tasks in large-scale systems will lead to

- excessive burdensome computational workload for the specific central computation node and chaos throughout the entire system when the central node is affected,
- connection and transmission problems when some sensors are located far away from the central computational node,
- scalability problems since the upper bound of the computation power is limited by the capacity of the central server.

To overcome the drawbacks of centralized methods, decentralized computations are considered to apply for FD [22, 75]. However, for these methods, each sensor node only uses local measurements to achieve FD and thus the performance of FD is limited when compared with centralized methods.

1.2 Motivation and Objective

With the development in sensing hardware and communication techniques, a diversity of high computational performance sensor networks are applied in modern industrial areas. A sensor network is a group of sensors and each sensor monitors processes and collects data in a different and limited location, and commutes the data with other sensors through a communication network. This property provides a possibility to combine advanced FD approaches and sensor networks for more effective, flexible, and scalable distributed methods when compared with centralized and decentralized ones. In the last decades, intensive attention has been paid to distributed FD strategies, where each sensor node can

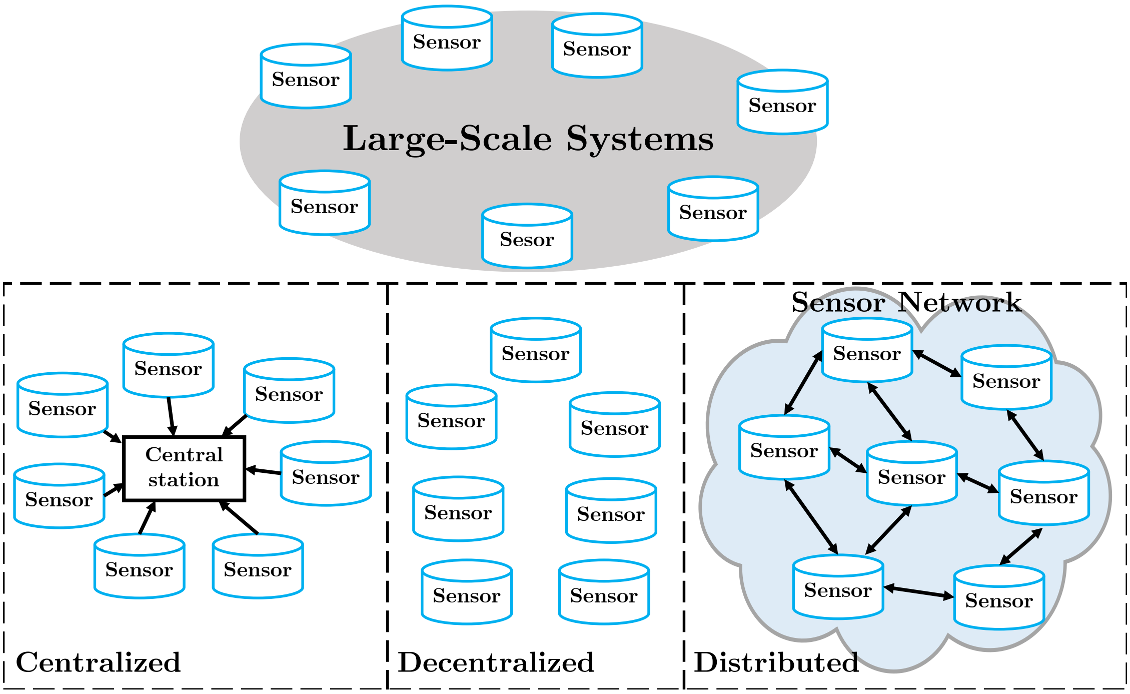


Figure 1.4: Centralized, decentralized and distributed methods

itself execute the FD algorithms based on the local measurement data as well as the data received via networks from other sensors. One possibility is the local residual generator, it allows each node to execute the FD algorithm based on the local measurement and also the neighbours' information; however, the design of distributed observer gain is a challenging task in such a structure. Sufficient conditions and numerical solutions are provided in [121, 122, 123] to reduce the design complexity. And applications of such structure to handle distributed fault isolation are also provided in [10, 39, 126]. Besides local residual generators, distributed average consensus algorithms [113], which compute the global average of sensor data in a distributed fashion, are also applied to perform distributed FD. Based on consensus and diffusion, distributed sensor FD is proposed in [66, 73]. And consensus based distributed realization of Kalman-filter and H_2 filter based FD are introduced in [27] and [119]. They achieve a specified FD performance while limiting the complexity of the algorithm. Another category of distributed algorithms is based on multivariate analysis (MVA) methods. For example, methods based on CCA [19, 61] aim at using correlation information from neighbours to reduce uncertainties and improve the performance of FD. And methods based on PCA [62, 130] consider fault-relevant variable selection and use Bayesian inference to propose efficient distributed FD. Most recently, machine learning methods are also involved into distributed design of FD methods, a distributed FD method based on deterministic learning is proposed in [17] and a combination of auto-encoder, SVM and fuzzy deep NN is applied in [60] to detect sensor fault with limited computation, memory, and energy resources of the sensors. Despite the existing distributed methods, many of them are limited to numerical and sufficient conditions or specific to certain systems that require assumptions. Moreover, there remains a significant challenge in designing optimal or suboptimal distributed FD systems that address the convergence time, network transmission time, and cost-effectiveness.

Based on the aforementioned observations, this work focuses on implementing model-based distributed FD algorithms for large-scale and interconnected systems. To be more specific, the goals of this study are:

- to realize the optimal centralized FD method in a distributed manner based on distributed average consensus;
- to propose a sub-optimal distributed FD method using distributed observer and post-filter to achieve a certain performance index based on linear matrix inequality (LMI);
- to provide an optimal distributed FD solution using one-step prediction, filtering, and smoothing.

Also, the industrial application is the other goal in addition to the theoretical contributions. In the benchmark examples, the effectiveness and applicability of the suggested methods are shown.

1.3 Outline of the Thesis

This thesis consists of eight chapters, which are structured as shown in Figure 1.5. The major objectives and contributions of each chapter are briefly summarized as follows.

Chapter 1: Introduction

This chapter presents the motivations, objectives, contributions, and organization of this thesis.

Chapter 2: Useful Knowledge and Preliminaries

This chapter introduces the fundamental knowledge about least squares (LS) estimation, graph theory, distributed average consensus, and LMI technologies. LS algorithm is widely applied in estimation area and builds the fundamental for optimal FD both in static and dynamic cases. Graph theory is used to describe the communication network to set the logic of communication between subsystems for distributed computation, such as average consensus. The distributed average consensus technology is considered as a powerful tool to handle the distributed coordination and network synchronization, and thus applied in this study. And LMI technology emerges as a numerical method to approach complicated design issues.

Chapter 3: The Basics of Fault Detection Technologies

This chapter is devoted to the overview of FD methodologies for static and dynamic systems influenced by stochastic noise and deterministic disturbance. These fundamental methodologies serve as the basis for the subsequent studies. First, the model of static systems and FD methods for them are reviewed. Then the model of dynamic systems and the study on model-based residual generation are introduced. Different types of residual generator stand for an analytical redundancy of the system under monitoring and build a link between static and dynamic models. The residual signals, which are the output of residual generator, are then used to detect faults in dynamic systems. Finally, a type of data-driven method is also introduced.

Chapter 4: Distributed Fault Detection in Large-Scale Systems Based on Distributed Average Consensus

The main objective of this chapter is to develop an optimal distributed FD approach for large-scale systems in the presence of unknown deterministic disturbances using the measurement of sensor networks. The design approach consists of two phases: the distributed offline training phase and the online implementation phase. Both phases are realized based on distributed iterative computation and average consensus algorithm. This approach is considered as a distributed realization of the centralized optimal method, which is based on $\mathcal{H}_2$ observer.

Chapter 5: Distributed Fault Detection in Interconnected Systems Based on Linear Matrix Inequality

In this chapter, a distributed FD method is designed for large-scale and interconnected systems. Each subsystem is equipped with a local observer-based residual generator. It uses only its local and neighbours' information to estimate their local states and then produces residual signal to detect fault. Also, a post-filter is applied to enhance the influence of fault to residual signal and meanwhile to reduce the influence of disturbance to residual signal. The post-filter is designed using a combination of PSA and DO.

Chapter 6: Distributed Fault Detection in Interconnected Systems via Optimal Estimation

This chapter presents distributed approaches to solve the problem of FD for interconnected systems by taking into account the influence of noise and transmission time of information exchange. For static cases, it uses the correlation between different signals to reduce the uncertainty for FD. For dynamic cases, firstly, the one-step prediction based on the measured data is implemented in a distributed fashion, such that the corresponding estimations and the innovation sequences of each node can be received in real-time manner. Then the innovation sequences are applied to improve the estimation result delivered from the one-step prediction by filtering and smoothing. Finally, the distributed approach uses the estimation result to generate residual signals and detect faults.

Chapter 7: Benchmark Study

In this chapter, the methods proposed in this thesis are tested on benchmark processes. The distributed realization of optimal FD proposed in Chapter 4 is applied for DC micro-grid. And the FD methods based on the distributed state observer and post-filter introduced in Chapter 5 is adopted for mass-spring system. Additionally, the effectiveness of the distributed FD method shown in Chapter 6 is demonstrated using the six-tank system.

Chapter 8: Conclusions and Future Works

This chapter concludes the thesis and discusses future work.

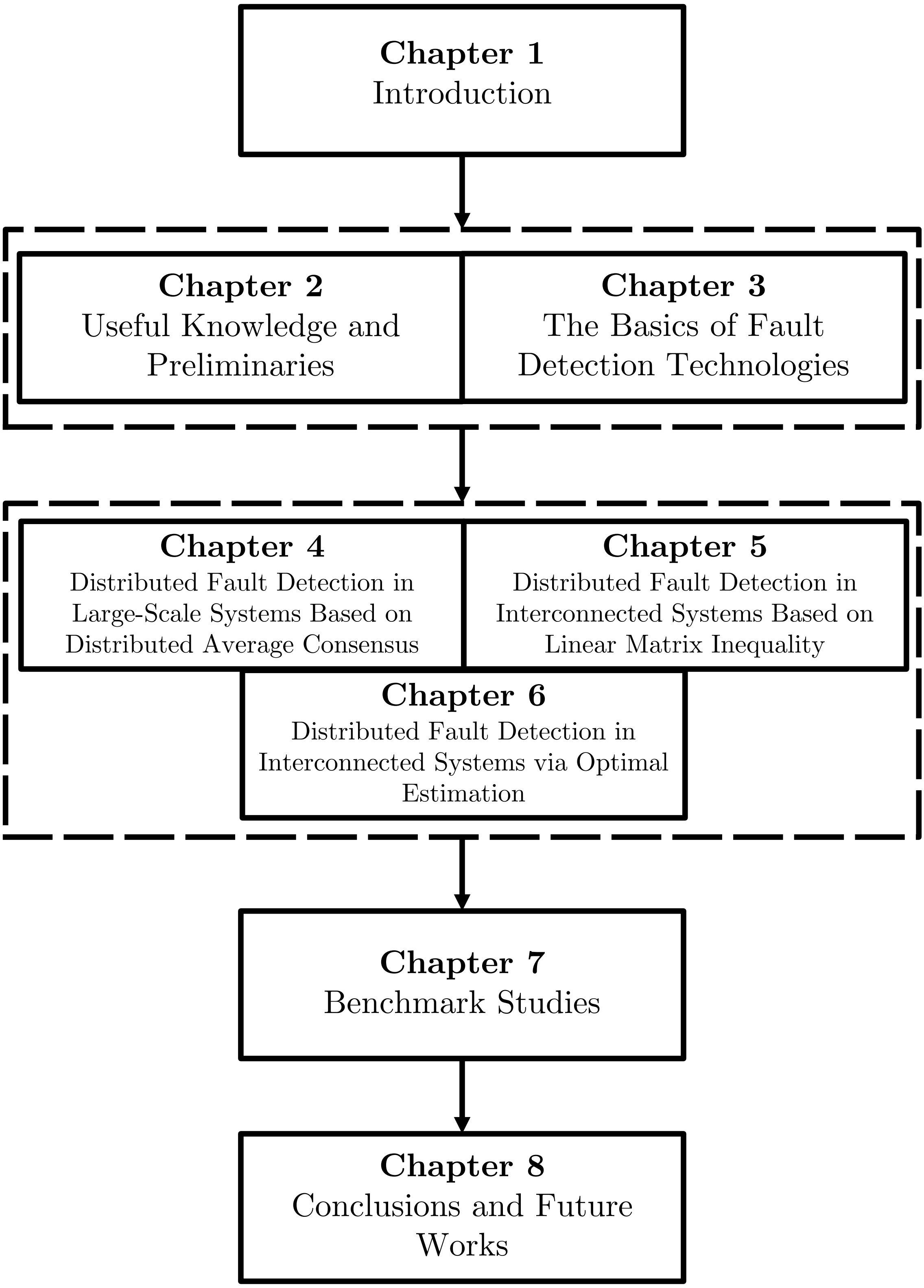

Chapter 1
Introduction
Chapter 2
Useful Knowledge and Preliminaries
Chapter 3
The Basics of Fault Detection Technologies
Chapter 4
Distributed Fault Detection in Large-Scale Systems Based on Distributed Average Consensus
Chapter 5
Distributed Fault Detection in Interconnected Systems Based on Linear Matrix Inequality
Chapter 6
Distributed Fault Detection in Interconnected Systems via Optimal Estimation
Chapter 7
Benchmark Studies
Chapter 8
Conclusions and Future Works

Figure 1.5: Organization of the chapters

2 Useful Knowledge and Preliminaries

This chapter provides a comprehensive overview of the basic concepts of LS estimation, graph theory, distributed average consensus, and LMI technology. Together with Chapter 3, these two chapters provide the reader with a comprehensive understanding of the distributed algorithms that will be discussed throughout the rest of the thesis.

2.1 Least Squares Estimation

The method of LS is about estimating variables by minimizing the squared discrepancies between observed data and their expected values. We first introduce concepts for the deterministic case.

Lemma 2.1. [64] *Given*

$$y = Hx + d, \tag{2.1}$$

where $x \in \mathcal{R}^n$ is the state variables, $y \in \mathcal{R}^m$ is measurement, $d \in \mathcal{R}^m$ is disturbance and rank$(H) = n$. The linear estimation

$$\hat{x} = (H^T W H)^{-1} H^T W y \tag{2.2}$$

delivers the optimal estimation of x in the sense of

$$\hat{x} = \arg\min_{x(y)} \|y - Hx(y)\|_W, \tag{2.3}$$

where $W > 0$ and $x(y)$ stands for arbitrary estimation of x given y.

Set $\hat{y} = H\hat{x}$ and estimation error $e_y = y - \hat{y}$. It is straightforward that

$$\langle e_y, H \rangle_W = 0, \tag{2.4}$$

where $\langle a, b \rangle_W = a^T W b$ stands for the weighted inner product. Denoting the column space of matrix H by $\mathcal{R}(H)$, equation (2.4) means that $\hat{y}$ is realized by geometrically projecting

y onto the space $\mathcal{R}(H)$ and the estimation error e_y is orthogonal to the space $\mathcal{R}(H)$ through weighted inner product with factor W. And for stochastic cases, we have a similar formulation as shown in the following.

Lemma 2.2. [4] *Given*

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right) \tag{2.5}$$

and measurement y, then

$$\hat{x} = \Sigma_{xy}\Sigma_y^{-1}y \tag{2.6}$$

delivers the optimal estimation of x in the sense of

$$\hat{x} = \arg\min_{x(y)} \operatorname{var}(x - x(y)), \tag{2.7}$$

where $x(y)$ are any arbitrary estimation of x given y.

Set the estimation error $e_x = x - \hat{x}$. From (2.5) and (2.6),

$$\begin{bmatrix} e_x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx} & 0 \\ 0 & \Sigma_y \end{bmatrix}\right). \tag{2.8}$$

From (2.8), it holds that

$$\operatorname{cov}(e_x, y) = \mathrm{E}(e_x y^T) = 0. \tag{2.9}$$

With the property of the trace function [127] and (2.9), we have

$$\mathrm{E}(\langle e_x, y\rangle) = \operatorname{tr}(\mathrm{E}(e_x y^T)) = 0. \tag{2.10}$$

Set $\mathcal{L}(y)$ as the linear space spanned by y. From (2.10), estimation $\hat{x}$ can be understood as the orthogonal projection of x onto the linear subspace $\mathcal{L}(y)$ and the estimation error e_x is orthogonal to $\mathcal{L}(y)$. Since random variables x and y in (2.5) are zero-mean variables, the orthogonality (2.10) also means e_x and y are uncorrelated. For general random variables with non-zero means

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_x \\ m_y \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right), \tag{2.11}$$

we remove means by $x_c = x - m_x$ and $y_c = y - m_y$ to guarantee that x_c and y_c satisfy the condition (2.5) and it holds that

$$\hat{x} = m_x + \Sigma_{xy}\Sigma_y^{-1}(y - m_y) \tag{2.12}$$

is a direct extension to Lemma 2.2.

From the above observations, it can be seen that the applications of the LS method in both deterministic and stochastic frameworks exhibit both differences and similarities. These properties are summarized in table 2.1.

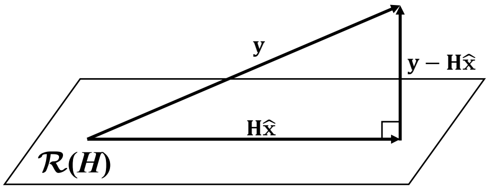
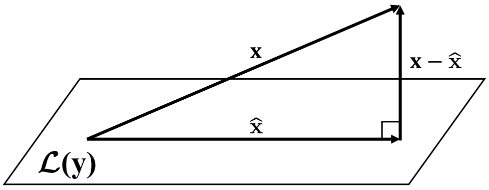
Deterministic Framework	Stochastic Framework
Model	
$y = Hx + d$	$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}\right)$
Problem	
$\min\limits_{x(y)} \lVert y - Hx(y) \rVert_W$	$\min\limits_{x(y)} \operatorname{var}(x - x(y))$
Solution	
$\hat{x} = (H^T W H)^{-1} H^T W y$	$\hat{x} = \Sigma_{xy} \Sigma_y^{-1} y$
Geometric Explanation	
	

Table 2.1: Deterministic and stochastic frameworks

2.2 Graph Theory

A graph is a representation of a set of nodes, the way how they are connected, and their metrical properties. In this study, it is applied to configure the physical structure of large-scale systems and the communication topology of sensor networks. Commonly, a graph $\mathcal{G}$ consisted of N nodes is denoted by

$$\mathcal{G} = (\mathcal{N}, \mathcal{E}). \tag{2.13}$$

Set $\mathcal{N} = \{1 \cdots, N\}$ is a finite nonempty node set. An edge (i, j) with $i, j \in \mathcal{N}$ is a link to connect the nodes i and j. In directed graphs, all the edges are directed from one node to another, i.e., (i, j) and (i, j) are different. In contrast, a graph where the edges are bidirectional is called an undirected graph. And $\mathcal{E} \subset N \times N$ is the edge set. The set of neighbours of node i in a graph, which is denoted by $\mathcal{N}_i = \{j|\ (i,j) \in \mathcal{E}\}$, is the set of nodes that have an edge between node i and themselves. A path in a graph is a sequence of edges and also can be considered as a sequence of nodes with the property that each node in the sequence is connected to the node by an edge next to it. A path that does not repeat nodes is called a simple path. The number of edges included in a path is the length of the path. An undirected graph is connected if there is a simple path between every pair of distinct nodes in this graph. A loop is an edge that connects a node to itself. If a graph has more than one edge joining some pair of nodes then these edges are called multiple edges. Figure 2.1 shows examples of loop and multiple edges. A simple graph is a graph that does not have loops and multiple edges. In the rest of this thesis and corresponding to our application, if no additional interpretation is mentioned, the graph is undirected, connected, and simple, each edge in the graph is an unordered and ascending pair of distinct nodes and each path is a simple path. An example of such graphs with algebraic description

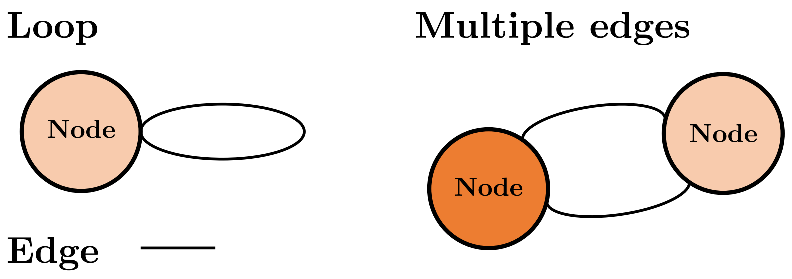


Figure 2.1: Loop and multiple edges

$$\mathcal{G} = (\mathcal{N}, \mathcal{E}),\ \mathcal{N} = \{1,2,3,4,5\},\ \mathcal{E} = \{(1,2),(2,3),(2,4),(3,5),(4,5)\}$$

is shown in Figure 2.2.

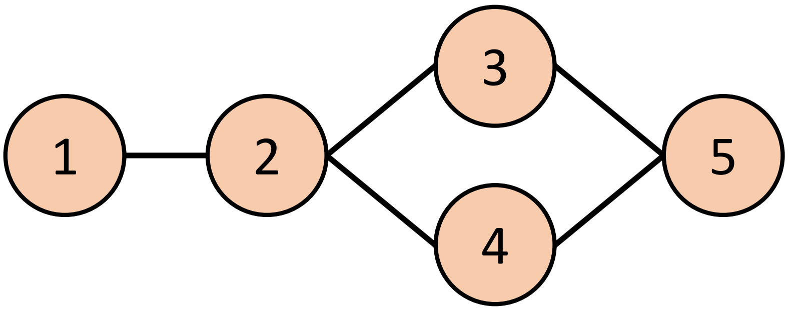


Figure 2.2: A simple, undirected and connected graph

Given a simple and undirected graph $\mathcal{G}$ with N nodes, the degree matrix $D = [d_{ij}] \in \mathcal{R}^{N\times N}$ of $\mathcal{G}$ is a diagonal matrix with

$$d_{ij} = \begin{cases} d_i, & i = j \\ 0, & i \neq j \end{cases}, \tag{2.14}$$

where d_i is the degree of node i, that is, the number of edges attached to node i. It is obvious that $d_i = \text{card}(\mathcal{N}_i)$. The adjacency matrix $A = [a_{ij}] \in \mathcal{R}^{N\times N}$ of $\mathcal{G}$ is a symmetric (0,1)-matrix with zeros on its diagonal, whose eigenvalues and eigenvectors are used to analyse the graph in spectral graph theory. The off diagonal elements of the adjacency matrix are defined by

$$a_{ij} = \begin{cases} 1, & (i,j) \in \mathcal{E} \\ 0, & (i,j) \notin \mathcal{E} \end{cases}, \tag{2.15}$$

and used to indicate whether pairs of nodes are connected or not. With the definition of degree matrix D and adjacency matrix A, the Laplacian matrix $L = [l_{ij}] \in \mathcal{R}^{N\times N}$ is defined by

$$L = D - A.$$

For the graph in Figure 2.2, the corresponding matrices are

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}, L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$

Graph theory has been applied in a wide range of fields, including communication networks, data organization, and formation control schemes, among others. In the following, we briefly introduce one useful property of the Laplacian matrix and based on it, a popular application in the area of formation control.

Lemma 2.3. [48] *Given a simple, undirected and connected graph $\mathcal{G}$ with N nodes, its Laplacian matrix $L = [l_{ij}] \in \mathcal{R}^{N\times N}$ is positive semidefinite.*

Lemma 2.4. [92] *Given a simple, undirected and connected graph $\mathcal{G}$ with N nodes and its Laplacian matrix $L = [l_{ij}] \in \mathcal{R}^{N\times N}$. For*

$$\dot{x}(t) = -Lx(t), x(0) = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^T, \tag{2.16}$$

we have

$$\lim_{t\to\infty} x(t) = \begin{bmatrix} \bar{x} & \cdots & \bar{x} \end{bmatrix}^T, \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{2.17}$$

Lemma 2.4 is the application of graph theory in the area of formation control, it is also called distributed average consensus [92]. By applying (2.16), the states of all nodes will converge to a common value, which is the average value of all initial states. A simulation result of Lemma 2.4 applied in the graph described in Figure 2.2 is shown in Figure 2.3.

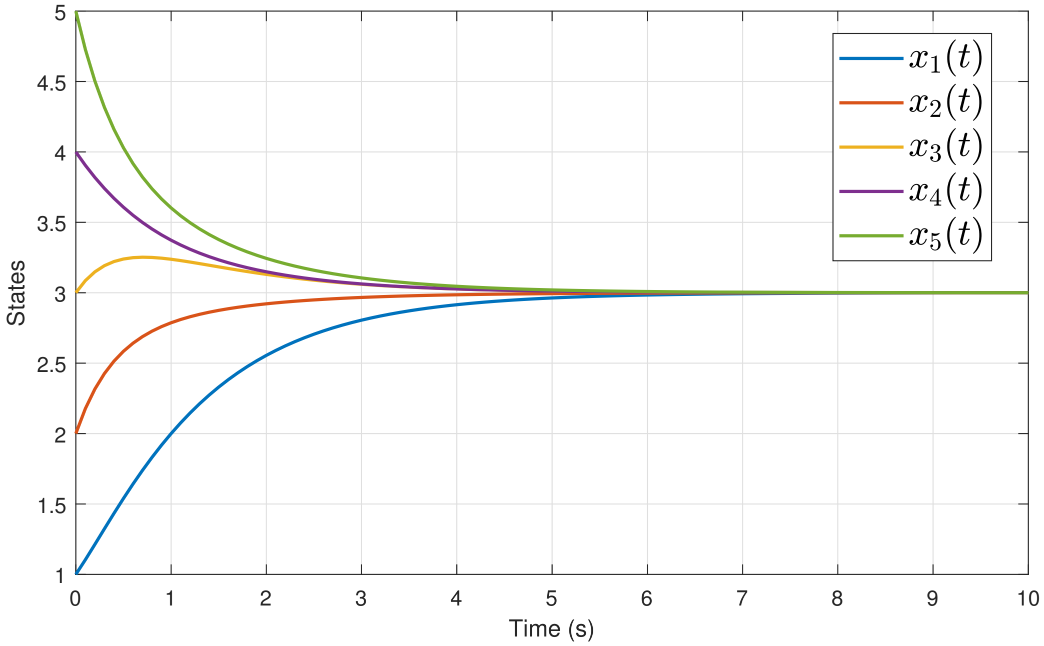


Figure 2.3: Simulation of formation control

2.3 Distributed Average Consensus

Distributed average consensus, which is applied in load balancing [23], distributed formation control [13] and distributed state estimation [86], is an important algorithm for large-scaled and distributed computing. Lemma 2.4 provides an example of a consensus algorithm in continuous case. In this section and the rest of this study, we mainly consider this algorithm in discrete form. Given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with N nodes, by distributed average consensus algorithm, each node has initial values and updates its value with a weighted average of local and its neighbors values. That is, the average consensus algorithm is achieved by performing

$$x_i(k+1) = w_{ii}x_i(k) + \sum_{j \in \mathcal{N}_i} w_{ij}x_j(k), \tag{2.18}$$

where $i \in \mathcal{N}$ and $x_i(0)$ is the initial values. Equation (2.18) can be modified as

$$x(k+1) = Wx(k), \tag{2.19}$$

where $x^T = \begin{bmatrix} x_1^T & \cdots & x_N^T \end{bmatrix}^T$ and $W = [w_{ij}] \in \mathcal{R}^{N \times N}$. By construction, the matrix W satisfies

$$W \in \mathcal{S},\ \mathcal{S} = \{W | w_{ij} = 0 \text{ if } i \neq j \text{ and } (i,j) \notin \mathcal{E}\}\,. \tag{2.20}$$

To achieve the average consensus, we must have

$$\lim_{k \to \infty} x(k) = \lim_{k \to \infty} W^k x(0) = \frac{1}{n}\mathbf{1}\mathbf{1}^\top x(0) \tag{2.21}$$

for any initial condition $x(0)$, where $\mathbf{1}$ is a vector with all element 1, or equivalently

$$\lim_{k \to \infty} W^k = \frac{1}{n}\mathbf{1}\mathbf{1}^\top. \tag{2.22}$$

Lemma 2.5. [112] *The equation* (2.22) *holds if and only if*

$$\begin{aligned} W\mathbf{1} &= \mathbf{1}, \\ \mathbf{1}^\top W &= \mathbf{1}^\top, \\ \rho\left(W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right) &\leqslant 1. \end{aligned}$$

Numerous methods have been proposed to solve W, which yield Lemma 2.5. For example, the max-degree weights method [113] calculates W as

$$w_{ij} = \begin{cases} \dfrac{1}{d} & j \in \mathcal{N}_i \\ 0 & j \notin \mathcal{N}_i \text{ and } j \neq i\,. \\ 1 - \dfrac{d_i}{d+1} & i = j \end{cases} \tag{2.23}$$

We can also apply the Metropolis-Hastings weights method [113], which can be calculated in a distributed manner as

$$w_{ij} = \begin{cases} \dfrac{1}{\max\{d_i, d_j\} + 1} & j \in \mathcal{N}_i \\ 0 & j \notin \mathcal{N}_i \text{ and } j \neq i\,. \\ 1 - \sum\limits_{q \neq i} w_{iq} & i = j \end{cases} \tag{2.24}$$

For example, given graph in Figure 2.2, use max-degree weights to calculates W, we have

$$W = \begin{bmatrix} 0.7500 & 0.2500 & 0 & 0 & 0 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 & 0 \\ 0 & 0.2500 & 0.5000 & 0 & 0.2500 \\ 0 & 0.2500 & 0 & 0.5000 & 0.2500 \\ 0 & 0 & 0.2500 & 0.2500 & 0.5000 \end{bmatrix}. \tag{2.25}$$

A simulation of (2.19) using W in (2.25) is shown in Figure 2.4.

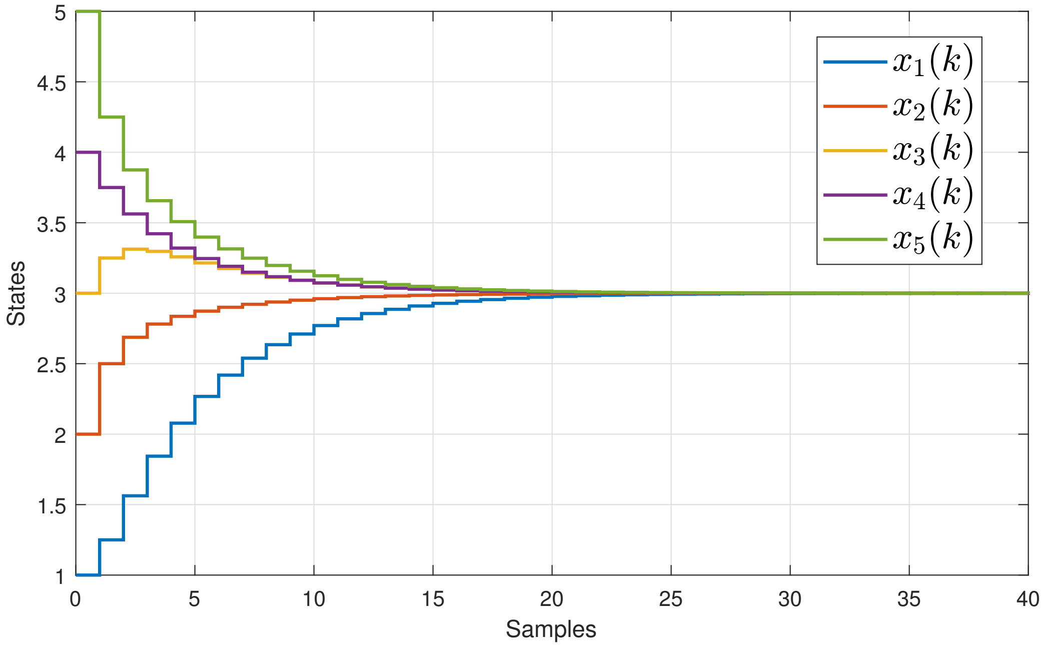


Figure 2.4: Simulation of distributed average consensus

2.4 Linear Matrix Inequality

LMIs are a regular occurrence in applications involving systems and control theory. Numerous analytical and synthesis issues in these domains can be resolved as optimization or feasibility problems using LMI technique. The LMIs have an expression of the form

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i < 0 \tag{2.26}$$

where

- the real number x_i is the decision variable and $x = (x_1, \cdots, x_m)$ is collection of all decision variables,

- $F_i \in \mathcal{R}^{n \times n}$, $i = 0, \cdots, m$, are given symmetric matrices,

- < 0 stands for negative definite, i.e., the largest eigenvalue of $F(x)$ is negative.

There are basically three kinds of problems that can be solved by the LMI technique.

- Feasibility:

$$\begin{aligned} \min_{t,x} \quad & t \\ \text{s.t.} \quad & F(x) \leqslant tI \\ & t \leqslant 0 \end{aligned}$$

- Linear objective minimization:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & F(x) \leqslant 0 \end{aligned}$$

- Generalized eigenvalue minimization:

$$\begin{aligned} \min_{\lambda,x} \quad & \lambda \\ \text{s.t.} \quad & F_1(x) < \lambda F_2(x) \\ & 0 < F_2(x) \\ & F_3(x) < 0 \end{aligned}$$

In system and control theories, some design issues are transferred into these three LMI problems. Then the problem can be numerically solved in MATLAB by using the LMI Toolbox [42] or YALMIP [76]. In the following, we briefly introduce two useful properties, which will be used later.

Lemma 2.6 (Schur Complement Lemma)**.** [29] *Given matrix*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{12} = A_{21}^{\mathrm{T}}$. *Then* $A > 0$ *if and only if*

$$A_{11} > 0 \quad \textit{and} \quad A_{22} - A_{21}A_{11}^{-1}A_{12} > 0$$

or

$$A_{22} > 0 \quad \textit{and} \quad A_{11} - A_{12}A_{22}^{-1}A_{21} > 0.$$

Similarly, $A < 0$ *if and only if*

$$A_{11} < 0 \quad \textit{and} \quad A_{22} - A_{21}A_{11}^{-1}A_{12} < 0$$

or

$$A_{22} < 0 \quad \textit{and} \quad A_{11} - A_{12}A_{22}^{-1}A_{21} < 0.$$

Lemma 2.7. [90] *Matrix* A *is Schur matrix if and only if*

$$A^{\mathrm{T}}PA - P < 0, \;\; P > 0. \tag{2.27}$$

Or with an additional matrix variable X*,*

$$\begin{bmatrix} -P & A^T X \\ X^T A & -X - X^T + P \end{bmatrix} < 0, \;\; P > 0. \tag{2.28}$$

2.5 Concluding Remarks

The basics of LS estimation, graph theory, distributed average consensus and LMI are introduced in this chapter. Notice that we only introduce the corresponding parts of each area which are related to our study. For more detailed properties of LS estimation, readers are referred to [4, 64]. The book [110] gives a comprehensive introduction to graph theory. Different concepts and schemes of designing a weighting matrix in (2.19) are proposed in [112, 113, 114]. The application of distributed average consensus in formation control is widely discussed in [92, 93]. A state of the art of theories, usages, and applications of LMI in the broad field of systems and control are studied in [12, 98]. Also, applications of LMI in FD are published in [75, 106].

3 The Basics of Fault Detection Technologies

In this chapter, an overview of the state of the art to topics of FD techniques [24, 25, 27] is presented. The purpose of this chapter is to give the fundamental knowledge to topics of FD and builds the foundation for further study. In the first section, optimal FD problems are formulated both in probabilistic and deterministic frameworks. Then, the mathematical descriptions of static systems as well as FD solutions for them are given. The third section focuses on dynamic systems, including modelling methods and model-based residual generator schemes for dynamic systems. Among different types of residual generators, Kalman filter-based and $\mathcal{H}_2$ observer-based one are applied to perform optimal FD in dynamic systems influenced by random noises and deterministic disturbances, respectively. Finally, a data-driven scheme is introduced to identify essential parameters, which are then used to construct DO for online FD.

3.1 Fault Detection Problems

A basic FD system is achieved by designing evaluation function J and threshold J_{th} to detect possible faults with decision logic

$$\begin{cases} J < J_{th} & , \quad \text{fault-free} \\ J \geqslant J_{th} & , \quad \text{faulty} \end{cases} . \tag{3.1}$$

Assume that the signal vector f is used to model the fault to be detected and satisfies

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

The evaluation function J is a mapping from measurements of the system to the feature of fault. Since the systems under consideration are influenced by unknown inputs, evaluation function J and threshold J_{th} are designed to fulfill certain indices, which are used to assess the performance of FD system.

3.1.1 Fault Detection in the Probabilistic Framework

In the probabilistic framework, false alarm rate (FAR), missed detection rate (MDR), and fault detection rate (FDR) are frequently used to assess FD performance in processes influenced by unknown inputs. These definitions are given below [27].

Definition 3.1. *Given the evaluation function J, threshold J_{th} and detection logic (3.1), FAR is defined as the probability*

$$\mathrm{FAR} = \Pr\left(J \geqslant J_{th} | f = 0\right).$$

Definition 3.2. *Given the evaluation function J, threshold J_{th} and detection logic (3.1), MDR is defined as the probability*

$$\mathrm{MDR} = \Pr\left(J < J_{th} | f \neq 0\right).$$

Definition 3.3. *Given the evaluation function J, threshold J_{th} and detection logic (3.1), FDR is defined as the probability*

$$\mathrm{FDR} = \Pr\left(J \geqslant J_{th} | f \neq 0\right).$$

In hypothesis testing, false alarm is also called type **I** error and missed detection is called type **II** error as shown by [84]. And notice that

$$\mathrm{FDR} = 1 - \mathrm{MDR},$$

we use only FAR and MDR to assess the performance of FD system and formulate the optimal FD problem in the probabilistic framework as finding J and J_{th} such that

$$(J, J_{th}) = \arg\min_{J, J_{th}} \mathrm{MDR}, \text{ s.t. } \mathrm{FAR} \leqslant \alpha, \tag{3.2}$$

where α is the acceptable level of FAR.

3.1.2 Fault Detection in Deterministic Processes

Under deterministic cases, the problem is modelled by

$$y = \mathcal{M}_d(d) + \mathcal{M}_f(f), \tag{3.3}$$

where $\mathcal{M}_d$ is the mapping from D to Y and $\mathcal{M}_f$ is the mapping from F to Y. Symbols D, F and Y are domains of d, f and y, respectively. In deterministic processes, it is assumed that the energy of d is bounded with the mathematical description

$$\|d\|_2 \leqslant \delta_d. \tag{3.4}$$

Since the bounded norm condition (3.4) is not a statistic property, FAR and MDR are not appropriate to evaluate FD performance in deterministic processes. This motivates us to find alternative assessments for the performance of FD in deterministic processes. Define two subspaces

$$I_d = \{y_d \in Y \mid y_d = \mathcal{M}_d(d),\ d \in D,\ \|d\|_p \leqslant \delta_d\}, \tag{3.5}$$

$$U_f = \{f \in F \mid y_f = \mathcal{M}_f(f) \in I_d\} \tag{3.6}$$

and formulate the optimal FD problem in deterministic processes as finding J and J_{th} such that

$$\forall y \in I_d,\ f = 0,\ J < J_{th}, \tag{3.7}$$

$$\forall f \notin U_f,\ d = 0,\ J \geqslant J_{th}. \tag{3.8}$$

Subspace U_f is known as the set of undetectable faults since the influence of these faults on y can't be separated from I_d, which represents the set of influence of all possible norm-bounded d on y. If condition (3.7) is satisfied, FAR equals zero and (3.8) means all faults, which are not undetectable, are detected when d equals zero.

3.2 Basic Methods for Fault Detection in Static Processes

In this section, we will briefly discuss the FD problem of static processes influenced by unknown external inputs, which are categorized into stochastic noises and deterministic disturbances.

3.2.1 Fault Detection with Random Noises

We first address the statistical cases. Given the model of a static system as

$$y = E_f f + \varepsilon, \tag{3.9}$$

where $y \in \mathcal{R}^m$ represents the measurement vector, $\varepsilon \sim \mathcal{N}(0, \Sigma)$ is the measurement noise and it is assumed that $\Sigma > 0$, $f \in \mathcal{R}^{k_f}$ is the fault vector and E_f is the distribution matrix and satisfies

$$\operatorname{rank}(E_f) = k_f \leqslant m. \tag{3.10}$$

Condition (3.10) means that measurement y measures all information of f and can be used to recover f.

Theorem 3.1. [32] *Given model* (3.9),

$$\begin{gathered} J = y^T \left(E_f^-\right)^T \left(E_f^- \Sigma \left(E_f^-\right)^T\right)^{-1} E_f^- y \sim \chi^2(k_f),\ J_{th} = \chi^2_\alpha(k_f), \\ E_f^- = \left(E_f^T \Sigma^{-1} E_f\right)^{-1} E_f^T \Sigma^{-1} \end{gathered} \tag{3.11}$$

deliver the solution for optimal problem (3.2).

3.2.2 Fault Detection with Deterministic Disturbances

For the deterministic processes, given the model of a static system as

$$y = E_f f + E_d d, \tag{3.12}$$

where $y \in \mathcal{R}^m$ represents the measurement vector, $d \in \mathcal{R}^{k_d}$ is the deterministic disturbance and it is assumed that $\|d\| \leqslant \delta_d$. Distribution matrix E_f has the same assumption (3.10) and

$$\operatorname{rank}(E_d) = m, \tag{3.13}$$

which means that d influences each dimension of y. We modify (3.5) and (3.6) as

$$I_d = \left\{y_d \in \mathcal{R}^m \mid y_d = E_d d,\ d \in \mathcal{R}^{k_d},\ \|d\| \leqslant \delta_d\right\}, \tag{3.14}$$

$$U_f = \left\{f \in \mathcal{R}^{k_f} \mid y_f = E_f f \in I_d\right\}. \tag{3.15}$$

for static model (3.12).

Theorem 3.2. [32] *Given model* (3.12),

$$\begin{gathered} J = y^T \left(E_d E_d^T\right)^{-1} M \left(E_d E_d^T\right)^{-1} y,\ J_{th} = \delta_d, \\ M = E_f \left(E_f^T \left(E_d E_d^T\right)^{-1} E_f\right) E_f^T, \end{gathered} \tag{3.16}$$

deliver the solution for optimal problem (3.7) *and* (3.8).

3.3 Basic Methods for Fault Detection in Dynamic Processes

In this section, we address the FD problem of dynamic processes.

3.3.1 Models of Nominal Dynamic Systems

We refer to nominal systems as systems that are fault-free and disturbance-free. In our work, it is assumed that the nominal systems are discrete-time linear time invariant (LTI) systems, which provide a simple solution to modelling dynamic systems and are extensively utilized in research and application areas. There are numerous ways to characterize a discrete-time LTI system.

One of them is the transfer function matrix, that is,

$$y(z) = G(z)u(z). \tag{3.17}$$

In (3.17), transfer function matrix $G(z)$ is a proper real-rational matrix and describes the input−output relation of a dynamic system in the frequency domain, $u \in \mathcal{R}^l$ is input vector, $y \in \mathcal{R}^m$ is output vector and z is the complex variable of z-transform, which converts a discrete-time signal into a complex frequency-domain representation.

The next one is state space representation:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k),\ x(0) = x_0 \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{3.18}$$

with $x \in \mathcal{R}^n$, $u \in \mathcal{R}^l$, and $y \in \mathcal{R}^m$ denoting the state vector, input vector and the output vector of the system, respectively. Matrices A, B, C, and D are real constant matrices with appropriate dimensions. State space models are also considered as a realization of transfer function matrices with

$$G(z) = D + C(zI - A)^{-1}B. \tag{3.19}$$

For simplicity of notation, relationship (3.19) is denoted by

$$G(z) = (A, B, C, D). \tag{3.20}$$

We assume that (A, B, C, D) is minimal realization, which is controllable and observable, has minimal order, and has the same response characteristics as the original model $G(z)$.

Also, the factorization approach has been widely utilized in the area of control theory and can be used to describe dynamic behaviours. In particular, doubly co-prime factorization (DCF), which has tight connections with parametrization of stabilizing controllers and plays an important role in dealing with robust control problems [129], is the other way to present dynamical characters of systems.

Definition 3.4. *Two transfer function matrices $M(z)$, $N(z)$ in $\mathcal{RH}_\infty$ are said to be right co-prime over $\mathcal{RH}_\infty$ if there exist two transfer function matrices $X(z)$, $Y(z)$ in $\mathcal{RH}_\infty$ such that*

$$Y(z)N(z) + X(z)M(z) = \begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} = I.$$

Definition 3.5. *Two transfer function matrices $\hat{M}(z)$, $\hat{N}(z)$ in $\mathcal{RH}_\infty$ are said to be left co-prime over $\mathcal{RH}_\infty$ if there exist two transfer function matrices $\hat{X}(z)$, $\hat{Y}(z)$ in $\mathcal{RH}_\infty$ such that*

$$\hat{M}(z)\hat{X}(z) + \hat{N}(z)\hat{Y}(z) = \begin{bmatrix} \hat{M}(z) & \hat{N}(z) \end{bmatrix} \begin{bmatrix} \hat{X}(z) \\ \hat{Y}(z) \end{bmatrix} = I.$$

For each proper real-rational transfer function matrix

$$G(z) = (A, B, C, D)\,,$$

there exist eight transfer function matrices $M(z)$, $N(z)$, $X(z)$, $Y(z)$, $\hat{M}(z)$, $\hat{N}(z)$, $\hat{X}(z)$ and $\hat{Y}(z) \in \mathcal{RH}_\infty$ with the state space realization

$$\begin{array}{ll} M(z) = (A + BF, B, F, I), & \hat{M}(z) = (A - LC, -L, C, I), \\ N(z) = (A + BF, B, C + DF, D), & \hat{N}(z) = (A - LC, B - LD, C, D), \\ X(z) = (A - LC, -(B - LD), F, I), & \hat{X}(z) = (A + BF, L, C + DF, I), \\ Y(z) = (A - LC, -L, F, 0), & \hat{Y}(z) = (A + BF, -L, F, 0), \end{array} \tag{3.21}$$

where F and L are chosen such that $A + BF$ and $A - LC$ are Schur matrices. Moreover, it holds that

$$G(z) = N(z)M(z)^{-1} = \hat{M}(z)^{-1}\hat{N}(z), \tag{3.22}$$

$$\begin{bmatrix} X(z) & Y(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix} \begin{bmatrix} M(z) & -\hat{Y}(z) \\ N(z) & \hat{X}(z) \end{bmatrix} = I, \tag{3.23}$$

$$\begin{bmatrix} M(z) & -\hat{Y}(z) \\ N(z) & \hat{X}(z) \end{bmatrix} \begin{bmatrix} X(z) & Y(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix} = I. \tag{3.24}$$

From (3.23) and (3.24), it is obvious that $N(z)$ and $M(z)$ are right co-prime over $\mathcal{RH}_\infty$ and $\hat{N}(z)$ and $\hat{M}(z)$ are left co-prime over RH_∞. And (3.22) is called DCF of $G(z)$, since a right co-prime factorization (RCF) of $G(z)$ is

$$G(z) = N(z)M(z)^{-1}, \tag{3.25}$$

a left co-prime factorization (LCF) is defined by

$$G(z) = \hat{M}(z)^{-1}\hat{N}(z) \tag{3.26}$$

and RCF and LCF together constitute the DCF of $G(z)$. If $G(z)$ is stable, for simplicity, we can set $F = 0$, $L = 0$ and

$$M(z) = \hat{M}(z) = I,\; N(z) = \hat{N}(z) = G(z),\; X(z) = \hat{X}(z) = I,\; Y(z) = \hat{Y}(z) = 0.$$

3.3.2 Model-Based Residual Generation Schemes

In this section, we introduce three common model-based residual generation strategies, which are the realization of software redundancy and build the core of model-based FD. In general, they either use input and output data of the considered system to build the estimation of output and then generate residual signal r by comparing the real output and its estimation or directly build residual signal using input, output data, and dynamic relation between them as shown in Figure 3.1 . One important characteristic feature of residual generator is

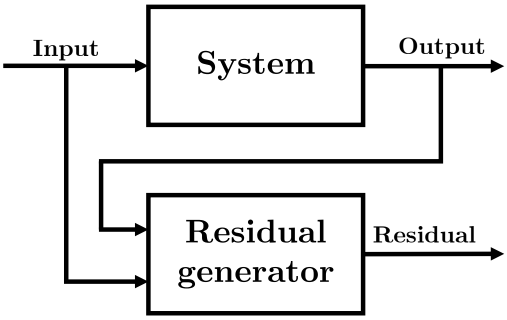


Figure 3.1: Residual generation

$$\forall u,\ x(0),\ \lim_{k\to\infty} r(k) = 0, \tag{3.27}$$

when the system has nominal behaviour.

3.3.2.1 Fault Detection Filter

One kind of observer-based residual generators is FDF, which was first proposed in [6] in the early 1970s. The foundation of FDF is a state observer:

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)), \\ \hat{y}(k) &= C\hat{x}(k) + Du(k), \end{aligned} \tag{3.28}$$

where $\hat{x} \in \mathcal{R}^n$ is the state vector of the observer and represents the estimation of x, $\hat{y} \in \mathcal{R}^m$ is the estimation of y, and L is the observer gain matrix. We define the residual signal as the difference between the real output and its estimation and can be simply written as

$$r(k) = y(k) - \hat{y}(k). \tag{3.29}$$

Further, introduce estimation error as

$$e(k) = x(k) - \hat{x}(k). \tag{3.30}$$

For a successful observation, it should hold that

$$\forall u,\ x(0),\ \lim_{k\to\infty} e(k) = 0 \tag{3.31}$$

in the fault-free and disturbance-free cases. It yields, by combing (3.18) and (3.28),

$$\begin{aligned} e(k+1) &= (A - LC)e(k), \\ \hat{y}(k) &= Ce(k). \end{aligned} \tag{3.32}$$

It is evident that choosing L such that $(A - LC)$ is Schur matrix guarantees (3.31). Notice that (3.31) is a sufficient condition for (3.27), then (3.27) is also guaranteed. From the above observation, we discover that choosing matrix L to achieve the state observer is the core part to design FDF. And in order to increase the degree of design freedom, we usually introduce matrix V as post-filter and change the residual signal into

$$r(k) = V\left(y(k) - \hat{y}(k)\right). \tag{3.33}$$

The above FDF scheme is based on a full-order observer, however, state estimation is not necessary for output estimation, which is required for FD. This motivates us to apply Luenberger type residual generator, which is also known as DO.

3.3.2.2 Diagnostic Observer

In this section, DO is introduced. It is one of the most extensively researched model-based residual generator forms because of its flexible structure and close relation with the Luenberger type observer. A DO is defined by

$$\begin{aligned} z(k+1) &= Gz(k) + Hu(k) + Ly(k), \\ r(k) &= Vy(k) - Wz(k) - Qu(k), \end{aligned} \tag{3.34}$$

where $z \in \mathcal{R}^s$ is the state of the observer and s is its order. The order s can be equal to, smaller, or larger than the system order n. Given model (3.18) and (3.34), the matrices G, H, L, V, W, Q together with $T \in \mathcal{R}^{s\times n}$ have to satisfy the so-called Luenberger conditions [87]:

$$\textbf{I}.\ G \text{ is Schur matrix} \tag{3.35}$$

$$\textbf{II}.\ TA - GT = LC,\ H = TB - LD \tag{3.36}$$

$$\textbf{III}.\ VC - WT = 0,\ Q = VD \tag{3.37}$$

Matrix T is used to define the estimation error

$$e(k) = Tx(k) - z(k). \tag{3.38}$$

If condition (3.35)−(3.37) are satisfied, it holds that

$$\begin{aligned} e(k+1) &= Ge(k), \\ r(k) &= Ve(k), \end{aligned} \tag{3.39}$$

in the fault-free and disturbance-free case and

$$\forall u, \ x(0), \ \lim_{k \to \infty} e(k) = 0. \tag{3.40}$$

It is straightforward that condition (3.27) is also fulfilled. Compared with the FDF scheme, it is clear that the DO scheme has a more flexible structure, which can lead to a reduced-order residual generator and can benefit online implementation. However, different from the FDF scheme, which has only two parameters L and V to be designed, the DO scheme has more parameters to be calculated and thus more involved effort for design.

3.3.2.3 Parity Space Approach

In this section, detection based on PSA is introduced. The PSA is a general framework for FD and was initiated by [21] in the early 1980s. Instead of using an observer, it constructs the residual generator based on the so-called parity relation. Defining parameters $\omega(k) \in \mathcal{R}^{\xi}$ and

$$\begin{aligned} \omega_s(k) &= \begin{bmatrix} \omega(k-s) \\ \vdots \\ \omega(k) \end{bmatrix} \in \mathcal{R}^{(s+1)\xi}, \ \Gamma_s = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix} \in \mathcal{R}^{(s+1)m \times n}, \\ H_{u,s} &= \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{s-1}B & \cdots & CB & D \end{bmatrix} \in \mathcal{R}^{(s+1)m \times (s+1)l}, \end{aligned} \tag{3.41}$$

where $\omega_s(k)$ is data structure to collecting data series from $\omega(k-s)$ to $\omega(k)$. Given model (3.18), ω can be u and y and matrices Γ_s and $H_{u,s}$ build the link between the original state space model (3.18) and the following model:

$$y_s(k) = \Gamma_s x(k-s) + H_{u,s} u_s(k). \tag{3.42}$$

Model (3.42) represents the relationship between the temporal input and output data in regard to the past state $x(k-s)$. It is stated in a simple manner, where all the parameters and matrices are known except $x(k-s)$. This motivates us to build residual signal as

$$r(k) = v_s \left(y_s(k) - H_{u,s} u_s(k) \right), \tag{3.43}$$

where $v_s \in \mathcal{R}^{(s+1)m}$ is called parity vector and if

$$v_s \Gamma_s = 0, \ v_s \neq 0, \tag{3.44}$$

the dynamics of $r(k)$ is governed by, in the fault-free and disturbance-free case,

$$r(k) = v_s \left(y_s(k) - H_{u,s} u_s(k) \right) = v_s \Gamma_s = 0. \tag{3.45}$$

Notice that the parity space is a space in which all elements are parity vectors [89], that is

$$P_s = \left\{ v_s \in \mathcal{R}^{(s+1)m} \mid v_s \Gamma_s = 0, \ v_s \neq 0 \right\}. \tag{3.46}$$

The simple solution to guarantee that space P_s is not empty is choosing $s \geqslant n$ such that

$$\operatorname{rank}(\Gamma_s) \leqslant n < (s+1)m. \tag{3.47}$$

Since matrix Γ_s is not full row rank, space P_s is not empty.

3.3.3 Fault Detection in Linear Stochastic Processes

Consider the process model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k), \ x(0) = x_0 \\ y(k) &= Cx(k) + Du(k) + v(k), \end{aligned} \tag{3.48}$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}^l$, and $y \in \mathcal{R}^m$ are introduced in model (3.18) and $w \in \mathcal{R}^n$ and $v \in \mathcal{R}^m$ denote the system and measurement noise, respectively. It is assumed that the noise signals $w(k)$ and $v(k)$ are white Gaussian processes and uncorrelated with input u and initial state vector $x(0)$.

$$x(0) \sim \mathcal{N}(x_0, P_0) \tag{3.49}$$

$$E\left(\begin{bmatrix} w(i) \\ v(i) \\ x(0) \end{bmatrix} \begin{bmatrix} w^\top(j) & v^\top(j) & x^\top(0) \end{bmatrix} \right) = \begin{bmatrix} \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \delta_{ij} & 0 \\ 0 & \Pi_0 \end{bmatrix}. \tag{3.50}$$

It is commonly known that a recursive Kalman filter with

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(k)(y(k) - \hat{y}(k)), \\ \hat{y}(k) &= C\hat{x}(k) + Du(k), \\ r(k) &= y(k) - \hat{y}(k), \\ \Sigma_r(k) &= CP(k \mid k-1)C^T + R, \\ L(k) &= (AP(k \mid k-1)C^T + S)\Sigma_r^{-1}(k), \\ P(k+1 \mid k) &= AP(k \mid k-1)A^T + Q - L(k)\Sigma_r(k)L^T(k), \ P(0|-1) = P_0 \end{aligned} \tag{3.51}$$

is a full-order observer (3.28) and delivers the best unbiased estimation of x in (3.48) since it minimizes the matrix P, which stands for the variance matrix of estimation error (3.30), and also matrix Σ_r, which is the variance matrix of white residual r . These properties enable us to solve the FD problem in a linear stochastic process at each instant in a manner that

$$r(k) = y(k) - \hat{y}(k) = f(k) + \varepsilon(k), \ \varepsilon(k) \sim \mathcal{N}\left(0, \Sigma_r(k)\right), \tag{3.52}$$

which is analogous to (3.9). And as introduced in Theorem 3.1,

$$J(k) = r^T(k)\Sigma_r^{-1}(k)r(k), \ J_{th} = \chi^2_\alpha(m) \tag{3.53}$$

delivers the optimal FD solution since for a given FAR α with minimizing the MDR. In a short summary, the Kalman filter based FDF (3.51) and residual evaluation (3.52) and (3.53) are optimal FD for linear stochastic process (3.48) because

- whiteness of the residual is essential for FD,
- variance matrix Σ_r in (3.52) is minimized by Kalman gain $L(k)$,
- for certain $L(k)$, residual evaluation (3.53) ensures that, for a given FAR α, the MDR is minimized.

3.3.4 Fault Detection in LTI Systems with Unknown Disturbance

Consider the process model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + E_d d(k) + E_f f(k), \ x(0) = x_0 \\ y(k) &= Cx(k) + Du(k) + F_d d(k) + F_f f(k), \end{aligned} \tag{3.54}$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}^l$, and $y \in \mathcal{R}^m$ are introduced in model (3.18) and $d \in \mathcal{R}^{k_d}$ is disturbance. It is assumed that

- $\|d\|_2^2 \leqslant \delta_d^2$
- $\operatorname{rank}(G_{yf}) = m$, $G_{yf} = (A, E_f, C, F_f)$
- $\forall\theta \in [0, 2\pi]$, $\begin{bmatrix} A - e^{j\theta}I & E_l \\ C_l & F_l \end{bmatrix}$ has full row rank

For purpose of optimal FD, we adopt the following $\mathcal{H}_2$ observer:

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)),\\ \hat{y}(k) &= Cx(k) + Du(k),\\ r(k) &= V(y(k) - \hat{y}(k)),\end{aligned}$$

where L and V are

$$\begin{aligned} R = CXC^T + F_dF_d^T,\ V = R^{-1/2},\\ L = (AXC^T + E_dF_d^T)R^{-1},\ X = AXA^T - LRL^T + E_dE_d^T. \end{aligned} \tag{3.55}$$

It is straightforward that the residual dynamics can be described by

$$\begin{aligned} r(z) &= V(G_d(z)d(z) + G_f(z)f(z)),\\ G_d(z) &= (A - LC, E_d - LF_d, C, F_d)\,,\\ G_f(z) &= (A - LC, E_f - LF_f, C, F_f)\,. \end{aligned} \tag{3.56}$$

The scheme (3.55) is optimal in the sense of

$$(L, V) = \arg\max_{K,R} \frac{\sigma_i\left(RN_f(e^{j\theta})\right)}{\|RN_d(z)\|_\infty},\ \forall\theta \in [0, 2\pi],\ i = 1, \cdots, m \tag{3.57}$$

$$N_d(z) = (A - KC, E_d - KF_d, C, F_d)\,, \tag{3.58}$$

$$N_f(z) = (A - KC, E_f - KF_f, C, F_f)\,. \tag{3.59}$$

The optimization problem means that L and V delivered by (3.55) can optimally balance the trade-off between the robustness of residual signal against disturbance and its sensitivity to fault [24]. Moreover, according to [129], the transfer function from disturbance to residual is co-inner, which means

$$VG_dG_d^T(z^{-1})V^T = I, \tag{3.60}$$

and it holds in the fault-free case

$$\|r(z)\|_2^2 = \|VG_d(z)d(z)\|_2^2 \leqslant \|d(z)\|_2^2. \tag{3.61}$$

In sense of property (3.60) and (3.61), as shown in [32], designing

$$J(k) = \|r\|_2,\ J_{th} = \delta_d \tag{3.62}$$

delivers the solution to optimal FD problem (3.7) and (3.8). In practice, the root mean square (RMS) value is often used instead of the $\mathcal{L}_2$ norm to form the evaluation function and threshold in (3.62). It is a measure of the average energy of a signal over a time interval and is defined as

$$J(k) = \|r\|_{RMS} = \sqrt{\frac{1}{N}\sum_{j=k-N+1}^{k} r^T(j)r(j)},\ J_{th} = \frac{\delta_d}{\sqrt{N}} \tag{3.63}$$

3.3.5 A Date-Driven Scheme

The above methods are model-based methods, which are based on the system model (3.18). When a model is unknown or difficult to build, data-driven methods are applied to design the residual generator. In this section, we introduce one type of data processing technique to do FD. First, introduce new notations

$$\begin{aligned} \Omega_k &= [\omega(k) \quad \cdots \quad \omega(k+N-1)] \in \mathcal{R}^{\xi \times N}, \\ \Omega_{k,s} &= [\omega_s(k) \quad \cdots \quad \omega_s(k+N-1)] \in \mathcal{R}^{(s+1)\xi \times N}, \\ H_{w,s} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{s-1} & \cdots & C & 0 \end{bmatrix} \in \mathcal{R}^{(s+1)m \times (s+1)n}, \end{aligned} \tag{3.64}$$

where $\omega(k)$, $\omega_s(k)$ is defined in (3.41) and N is a large integer, and rewrite model (3.48) into the following I/O data model

$$\begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix} = \Phi_s \begin{bmatrix} U_{k,s} \\ X_{k-s} \end{bmatrix} + \begin{bmatrix} 0 \\ H_{w,s}W_{k,s} + V_{k,s} \end{bmatrix}, \; \Phi_s = \begin{bmatrix} I & 0 \\ H_{u,s} & \Gamma_s \end{bmatrix}. \tag{3.65}$$

Since $\Phi_s \in \mathcal{R}^{(s+1)(m+l)\times(n+(s+1)l)}$ and for $s \geqslant n$, Φ_s is not full row rank and there exists $\Phi_s^\perp$ such that

$$\begin{aligned} \Phi_s^\perp \Phi_s = 0, \Phi_s^\perp \in \mathcal{R}^{((s+1)m-n)\times(s+1)(m+l)}, \\ \Phi_s^\perp \begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix} = \Phi_s^\perp \begin{bmatrix} 0 \\ H_{w,s}W_{k,s} + V_{k,s} \end{bmatrix}. \end{aligned} \tag{3.66}$$

It is straightforward that

$$r_s(k) = \Phi_s^\perp \begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix} \tag{3.67}$$

builds a residual generator since, in the fault-free and disturbance-free case, $r_s(k) = 0$. Next, we use a data-driven method to identify Φ_s. Let

$$Z_p = \begin{bmatrix} U_{k-s-1,s} \\ Y_{k-s-1,s} \end{bmatrix} \tag{3.68}$$

represents the past input and output data, which are uncorrelated with future noise $W_{k,s}$ and $V_{k,s}$, that is,

$$\frac{1}{N-1} W_{k,s} Z_p^T \approx 0, \; \frac{1}{N-1} V_{k,s} Z_p^T \approx 0. \tag{3.69}$$

Based on this observation, subspace identification methods (SIM) [30] are proposed to identify $\Phi_s^\perp$. In the following, we introduce a numerically reliable algorithm to identify $\Phi_s^\perp$ in a data-driven manner and then realize it by DO for online computation.

Algorithm 3.1. [25] *Data-driven scheme*

```
Offline Computation
Step 1  Select $s$, $N$ and form $Z_p$, $U_{k,s}$ and $Y_{k,s}$
Step 2  Do QR-decomposition
        $\begin{bmatrix} Z_p \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$
Step 3  Do singular value decomposition (SVD)
        $\begin{bmatrix} R_{21} & R_{22} \\ R_{31} & R_{32} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \ \Sigma_2 \approx 0$
Step 4  Select $\phi_s = [\beta_s \quad \alpha_s]$ be any row of $U_2^T$
        $\alpha_s = [\alpha_{s,0} \quad \cdots \quad \alpha_{s,s}] \in \mathcal{R}^{1\times(s+1)l}$, $\alpha_{s,i} \in \mathcal{R}^{1\times l}$, $i = 0, \cdots, s$
        $\beta_s = [\beta_{s,0} \quad \cdots \quad \beta_{s,s}] \in \mathcal{R}^{1\times(s+1)m}$, $\beta_{s,i} \in \mathcal{R}^{1\times m}$, $i = 0, \cdots, s$
Step 5  Construct parameters
        $G = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathcal{R}^{s\times s}, \ H = \begin{bmatrix} \beta_{s,0} \\ \beta_{s,1} \\ \vdots \\ \beta_{s,s-1} \end{bmatrix}, \ L = \begin{bmatrix} \alpha_{s,0} \\ \alpha_{s,1} \\ \vdots \\ \alpha_{s,s-1} \end{bmatrix}$
        $v = \alpha_{s,s}$, $w = [0 \quad \cdots \quad 0 \quad 1]$,$q = \beta_{s,s}$
Step 6  Construct DO
        $z(k+1) = Gz(k) + Hu(k) + Ly(k)$
            $r(k) = vy(k) - wz(k) - qu(k)$
Step 7  Run DO offline and compute the variance $\delta_r$ of $r(k)$
Step 8  Set $J_{th} = \chi^2_\alpha(1)$ for given FAR $\alpha$
Online Detection
Step 1  Run DO online and build $J(k) = r^2(k)/\delta_r$
Step 2  Detect fault using (3.1)
```

3.4 Concluding Remarks

This chapter presents the basics of FD technologies, including the preliminaries for both stochastic and deterministic systems. Since it is only a short introduction, many details are not involved. The first part focuses on the formulation of optimal FD problems for both stochastic and deterministic systems. Then the solution of optimal FD problems for static systems is introduced. A more detailed review of these parts is presented in [27]. It is followed by FD issues in dynamic systems. For dynamic systems, we start with different modelling methods, which are the basis of model-based FD. To realize the software redundancy, FDF, DO, and PSA are introduced to build residual generator for FD. Specifically, Kalman filter based and $\mathcal{H}_2$ observer based FDF are applied for dynamic systems to perform optimal FD in stochastic and deterministic systems, respectively. Here, Kalman filter and $\mathcal{H}_2$ observer are direct extensions of LS estimation introduced in Chapter 2, and the proof is shown in [64]. Other design procedures and detailed explanations about the model-based methods are introduced in [24]. Finally, a data-driven method is briefly introduced to identify an FD system and then transfer it into DO form. Interested readers are referred to [25, 26, 28] for the reviews of common and advanced data-driven methods of process monitoring and FD systems.

4 Distributed Fault Detection in Large-Scale Systems Based on Distributed Average Consensus

The aim of this chapter is to create a distributed version of the centralized optimal FD method for handling unknown deterministic disturbances in large-scale systems. We utilize a sensor network where each local sensor communicates with its neighbours to exchange information. The exchanged information is used for distributed offline training and also during online implementation. Both phases apply distributed average consensus and use iterative computation to guarantee the convergence of corresponding values to the same value achievable in the centralized optimal manner. To demonstrate the effectiveness of the proposed method, we present a simulation result and make a comparison between the performances of the distributed realization and the centralized optimal solution.

4.1 Preliminaries and Problem Formulation

In this section, we present an overview of the models for the dynamic system and sensor network, the centralized optimal FD scheme, and the problem formulation. These components form the foundation for further study.

4.1.1 Models description

In this chapter, we focus on a class of large-scale LTI systems, which are defined as follows:

$$x(k+1) = Ax(k) + E_p d_p(k) + E_f f(k), \tag{4.1}$$

where $x(k) \in \mathcal{R}^q$ denotes the state vector, $f(k) \in \mathcal{R}^f$ denotes the fault, $d_p(k) \in \mathcal{R}^p$ denotes the unknown deterministic disturbance in process, and A, E_p and E_f are known constant matrices with appropriate dimensions. For the purpose of FD and process monitoring, the system under consideration is equipped with a sensor network, which consists of n sensor nodes, and each sensor node is modelled by

$$y_i(k) = C_i x(k) + \bar{F}_i d_m(k), d_m = \begin{bmatrix} d_1^T & \cdots & d_n^T \end{bmatrix}^T \tag{4.2}$$

where $i \in \{1, \cdots, n\}$ denotes the index of sensor node, $y_i \in \mathcal{R}^{m_i}$ is the measurement of sensor node i, and C_i and $\bar{F}_i$ are known constant matrices with appropriate dimension, and d_i is the local measurement disturbance and d_m is the collection of measurement disturbances. Since the monitored parts of different sensors can overlap, some local measurement disturbances can influence more than one sensors. Thus, it is possible that

$$\bar{F}_i \bar{F}_j^T \neq 0. \tag{4.3}$$

The communication topology of the sensor network is described by a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N} = \{1, \cdots, n\}$ denotes the node set and $\mathcal{E} \subseteq \{\mathcal{N} \times \mathcal{N}\}$ stands for the edge set. Besides the notations introduced in Chapter 2, denote $d(i,j)$ as the minimal length of the paths connecting node i and j,

$$D_g = \max_{i,j \in \mathcal{N}} d(i,j)$$

as the diameter of $\mathcal{G}$, and $c_i = \text{card}(\mathcal{N}_i)$ as the cardinality of $\mathcal{N}_i$. Moreover, define matrix

$$\Sigma = \bar{F}_l \bar{F}_l^T = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_{n1} & \cdots & \Sigma_{nn} \end{bmatrix}, \ \bar{F}_l = \begin{bmatrix} \bar{F}_1 \\ \vdots \\ \bar{F}_n \end{bmatrix}, \ \Sigma_{ij} = \bar{F}_i \bar{F}_j^T. \tag{4.4}$$

In this chapter, the communication topology of the sensor network is first established based on the relationship between F_i and F_j, i.e., for $i, j \in \{1, \cdots, n\}$,

$$\begin{cases} j \in \mathcal{N}_i, & \Sigma_{ij} \neq 0 \text{ and } i \neq j, \\ j \notin \mathcal{N}_i, & \Sigma_{ij} = 0 \end{cases}. \tag{4.5}$$

Notice that (4.5) may not guarantee a connected graph. In this case, more links should be built to make the communication topology a connected one.

4.1.2 An Optimal Fault Detection Scheme

Stacking all sensor nodes together, the global models are given by

$$x(k+1) = Ax(k) + E_l d_l(k) + E_f f(k), \tag{4.6}$$

$$y_l(k) = C_l x(k) + F_l d_l(k), \tag{4.7}$$

where

$$y_l = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \; d_l = \begin{bmatrix} d_p \\ d_m \end{bmatrix}, \; C_l = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix}, \; E_l = \begin{bmatrix} E_p & 0 \end{bmatrix}, \; F_l = \begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix}, \; F_i = \begin{bmatrix} 0 & \bar{F}_i \end{bmatrix}$$

and

$$y_l \in \mathcal{R}^m, \; m = \sum_{i=1}^{n} m_i.$$

Since the global model (4.6) and (4.7) are the same as the model (3.54) introduced in Chapter 3, for detection purpose, we adopt the following observer-based residual generator

$$\hat{x}(k+1) = A\hat{x}(k) + L(y_l(k) - C_l\hat{x}(k)), \tag{4.8}$$

$$r(k) = V(y_l(k) - C_l\hat{x}(k)), \tag{4.9}$$

$$R = C_l X C_l^T + F_l F_l^T, \tag{4.10}$$

$$V = R^{-1/2}, \tag{4.11}$$

$$L = (AXC_l^T + E_l F_l^T)R^{-1}, \tag{4.12}$$

$$X = AXA^T - LRL^T + E_l E_l^T, \tag{4.13}$$

where $\hat{x}(k)$ represents the state estimation and $r(k)$ denotes the residual signal. Also, set

$$J(k) = \|r\|_{RMS} = \sqrt{\frac{1}{N}\sum_{j=k-N+1}^{k} r^T(j)r(j)}, \; J_{th} = \frac{\delta_d}{\sqrt{N}} \tag{4.14}$$

and use decision logic (3.1)

$$\begin{cases} J(k) < J_{th}, & \text{fault-free} \\ J(k) \geqslant J_{th}, & \text{faulty} \end{cases} .$$

for optimal FD. Notice that solutions (4.8)−(4.13) are directly from the optimal solution in Section 3.3.4. In order to ensure that discrete time algebraic Riccati equation (DARE) (4.13) has stabilizing solution, we assume

- $\text{rank}(G_{yf}) = m$, $G_{yf} = (A, E_f, C, F_f)$
- $\forall \theta \in [0, 2\pi]$, $\begin{bmatrix} A - e^{j\theta}I & E_l \\ C_l & F_l \end{bmatrix}$ has full row rank

Moreover, assume that F_l has full row rank which means that each dimension of y_l is influenced by disturbance. Based on this assumption, we have

$$\Sigma = F_l F_l^T$$

in (4.4) is a symmetric and positive definite matrix.

4.1.3 Problem Formulation

It is worth mentioning that to achieve the optimal FD approach given in (4.8)-(4.14), all the information about sensors and measurement data should be collected at one central station to perform the FD actions, which requires significant communication efforts for the central station. To deal with this issue, we investigate a distributed realization of the proposed optimal FD scheme, which delivers exactly the same optimal FD performance at each sensor node.

4.2 A Distributed Fault Detection Scheme

As mentioned above, in this section, a distributed realization of the proposed optimal FD scheme is presented, which is achieved by performing the following two phases:

- distributed offline training;
- distributed online FD.

We assume that at sensor node i, only A_l, E_l, C_i, F_i, and the local measurement y_i are available. Besides, the information can be transmitted between neighbours according to the communication topology (4.5).

4.2.1 Distributed Offline Training

It is clear that the most important part of the computation of parameters in (4.10)−(4.13) is to compute X in (4.13). For our purpose, (4.13) should be solved at each node in a distributed way. To deal with this issue, we define

$$I_1 = C_l^T \Sigma^{-1} C_l, \; I_2 = F_l^T \Sigma^{-1} C_l, \; I_3 = F_l^T \Sigma^{-1} F_l, \; \Omega = (X^{-1} + I_1), \tag{4.15}$$

and apply the matrix identity

$$R^{-1} = \Sigma^{-1} - \Sigma^{-1} C_l \Omega^{-1} C_l^T \Sigma^{-1}, \tag{4.16}$$

to reformulate (4.13) into

$$\begin{aligned} X =& AXA^T - LRL^T + E_l E_l^T \\ =& A \left(X - XC_l^T(\Sigma^{-1} - \Sigma^{-1} C_l \Omega^{-1} C_l^T \Sigma^{-1}) C_l X \right) A^T \\ & - AXC_l^T(\Sigma^{-1} - \Sigma^{-1} C_l \Omega^{-1} C_l^T \Sigma^{-1}) F_l E_l^T \\ & - E_l F_l^T(\Sigma^{-1} - \Sigma^{-1} C_l \Omega^{-1} C_l^T \Sigma^{-1}) C_l X A^T \\ & E_l \left(I - F_l^T(\Sigma^{-1} - \Sigma^{-1} C_l \Omega^{-1} C_l^T \Sigma^{-1}) F_l \right) E_l^T . \end{aligned} \tag{4.17}$$

Further, we have

$$
\begin{aligned}
&X - XC_l^T(\Sigma^{-1} - \Sigma^{-1}C_l\Omega^{-1}C_l^T\Sigma^{-1})C_lX \\
=&X - X(I - I_1(X^{-1} + I_1)^{-1})I_1X \\
=&X - X(I + I_1X^{-1})^{-1})I_1X = \Omega^{-1},
\end{aligned} \tag{4.18}
$$

$$
XC_l^T(\Sigma^{-1} - \Sigma^{-1}C_l\Omega^{-1}C_l^T\Sigma^{-1})F_l = X(I - C_l^T\Sigma^{-1}C_l\Omega^{-1})C_l^T\Sigma^{-1}F_l = \Omega^{-1}I_2^T, \tag{4.19}
$$

$$
I - F_l^T(\Sigma^{-1} - \Sigma^{-1}C_l\Omega^{-1}C_l^T\Sigma^{-1})F_l = I - I_3 + I_2\Omega^{-1}I_2^T, \tag{4.20}
$$

For distributed computation of X, apply Riccati recursion [68] with

$$
X(\eta+1) = AX(\eta)A^T - L(\eta)R(\eta)L(\eta)^T + E_lE_l^T,\ \lim_{\eta\to\infty} X(\eta) = X, \tag{4.21}
$$

$$
R(\eta) = C_lX(\eta)C_l^T + F_lF_l^T,\ L = (AX(\eta)C_l^T + E_lF_l^T)R(\eta)^{-1}. \tag{4.22}
$$

Combining (4.17)–(4.22), it holds that

$$
\begin{cases}
X(\eta+1) = & \Xi(\eta) - \Psi(\eta) - \Psi^T(\eta) + \Pi(\eta) \\
\Omega(\eta) = & (X^{-1}(\eta) + I_1) \\
\Xi(\eta) = & A\Omega^{-1}(\eta)A^T \\
\Psi(\eta) = & A\Omega^{-1}(\eta)I_2^TE_l^T \\
\Pi(\eta) = & E_l(I - I_3 + I_2\Omega^{-1}(\eta)I_2^T)E_l^T
\end{cases} \tag{4.23}
$$

provides a iterative way to compute X in (4.13) and

$$
\lim_{\eta\to\infty} X(\eta) = X,\ \lim_{\eta\to\infty} \Omega(\eta) = \Omega = (X^{-1} + I_1).
$$

It follows from (4.23) that, for iterative computation of X, I_1, I_2, and I_3 should be calculated first. Since they have a similar structure, we construct them as the unified form

$$
\Lambda^T\Sigma^{-1}\Phi = \begin{bmatrix}\Lambda_1^T & \cdots & \Lambda_n^T\end{bmatrix}\begin{bmatrix}\Sigma_{11} & \cdots & \Sigma_{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_{n1} & \cdots & \Sigma_{nn}\end{bmatrix}^{-1}\begin{bmatrix}\Phi_1 \\ \vdots \\ \Phi_n\end{bmatrix}, \tag{4.24}
$$

where Λ and Φ can be C_l or F_l, and Σ is the same as introduced in (4.4). We now introduce a new parameter Z as

$$
\Lambda^T\Sigma^{-1} = Z^T = \begin{bmatrix}Z_1^T & \cdots & Z_n^T\end{bmatrix} \tag{4.25}
$$

and divide the computation of $\Lambda^T\Sigma^{-1}\Phi$ into two phases: first $\Lambda^T\Sigma^{-1} = Z^T$ and further $Z^T\Phi$. After this partition, both parts can be calculated in a distributed way. Due to the reason that the first phase includes matrix inverse, which can cause huge computational costs, we apply a distributed iterative computation to avoid this problem and to solve the first computation. And for the second phase, the average consensus technique is implemented to let each sensor node obtain a common $\Lambda^T\Sigma^{-1}\Phi$. Finally, at the third phase, X is solved based on the results from the first two phases and (4.23).

$$\underbrace{\overbrace{\Lambda^T\Sigma^{-1}}^{\text{first phase}}\quad\Phi}_{\text{second phase}} = Z^T\Phi$$

$$\Downarrow\ I_1, I_2, I_3$$

$$\underbrace{X = A\Omega^{-1}A^T - A\Omega^{-1}I_2^TE_l^T - E_lI_2\Omega^{-1}A^T + E_l(I - I_3 + I_2\Omega^{-1}I_2^T)E_l^T}_{\text{third phase}}$$

4.2.1.1 First Phase

For calculating (4.25) and avoiding the inverse computation, we adopt the iterative computation method

$$Z(\zeta+1) = Z(\zeta) + \lambda(\Lambda - \Sigma Z(\zeta)). \tag{4.26}$$

The computation (4.26) is based on Richardson iteration [53] with ζ as the iteration number and λ being a constant factor, which is designed to guarantee the convergence

$$\lim_{\zeta\to\infty} Z(\zeta) = Z. \tag{4.27}$$

It is evident that equation (4.25) leads to

$$\Lambda = \Sigma Z \Longleftrightarrow \begin{bmatrix}\Lambda_1\\ \vdots\\ \Lambda_n\end{bmatrix} = \begin{bmatrix}\Sigma_{11} & \cdots & \Sigma_{1n}\\ \vdots & \ddots & \vdots\\ \Sigma_{n1} & \cdots & \Sigma_{nn}\end{bmatrix}\begin{bmatrix}Z_1\\ \vdots\\ Z_n\end{bmatrix}$$

Setting iteration error as $e(\zeta) = Z(\zeta) - Z$, the dynamic of $e(\zeta)$ is identified as

$$\begin{aligned} e(\zeta+1) &= Z(\zeta+1) - Z\\ &= Z(\zeta) + \lambda(\Sigma Z - \Sigma Z(\zeta)) - Z\\ &= (I - \lambda\Sigma)e(\zeta). \end{aligned} \tag{4.28}$$

It is clear that when $(I - \lambda\Sigma)$ is a Schur matrix, which means all eigenvalues of $(I - \lambda\Sigma)$ are located inside the unit disk,

$$\lim_{\zeta\to\infty} e(\zeta) = 0, \tag{4.29}$$

which also implies (4.27). Thus, $(I - \lambda\Sigma)$ being Schur matrix is the requirement to guarantee (4.27). In what follows, we introduce lemmas to provide the condition for λ to ensure that $(I - \lambda\Sigma)$ is Schur matrix.

Lemma 4.1. *If matrix S is real and symmetric, then*

$$\|S\|_2 \leqslant \|S\|_\infty.$$

Proof. From [49], it holds that $\|S\|_2 \leqslant \sqrt{\|S\|_1 \|S\|_\infty}$. If S is real and symmetric, we have $\|S\|_1 = \|S\|_\infty$, so $\|S\|_2 \leqslant \|S\|_\infty$. □

Lemma 4.2. *The condition*

$$0 < \lambda < \frac{2}{\|\Sigma\|_\infty} \tag{4.30}$$

can ensure that $(I - \lambda\Sigma)$ is Schur matrix when Σ is symmetric and positive definite matrix with all elements being real numbers.

Proof. Since Σ is a symmetric and positive definite matrix with all elements real numbers, we have $\lambda_{max}(\Sigma) = \|\Sigma\|_2$. Doing eigen decomposition of Σ leads to

$$\Sigma = U\Omega U^T,$$

where U is an orthogonal matrix and $U^{-1} = U^T$, and Ω is a diagonal matrix with diagonal elements being the eigenvalues of Σ. Further, we have

$$I - \lambda\Sigma = U(I - \lambda\Omega)U^T = U \begin{bmatrix} 1-\lambda\lambda_1 & & & 0 \\ & 1-\lambda\lambda_2 & & \\ & & \ddots & \\ 0 & & & 1-\lambda\lambda_n \end{bmatrix} U^T \tag{4.31}$$

where λ_i denotes the eigenvalue of Σ. Since Σ is positive definite, we have $\lambda_i > 0$. In order to ensure that $I - \lambda\Sigma$ is Schur matrix, from (4.31), it is clear that

$$0 < \lambda < \frac{2}{\lambda_{max}(\Sigma)}. \tag{4.32}$$

Moreover, according to Lemma 4.1,

$$\lambda_{max}(\Sigma) = \|\Sigma\|_2 \leqslant \|\Sigma\|_\infty. \tag{4.33}$$

If we combine (4.32) and (4.33), it is evident that Lemma 4.2 is proved. □

Based on Lemma 4.2, we apply (4.30) to compute λ. First, $\|\Sigma\|_\infty$ needs to be calculated in a distributed manner. For this purpose, we partition Σ into n rows as

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ \vdots \\ \Sigma_n \end{bmatrix} = \begin{bmatrix} F_1F_1^T & \cdots & F_1F_n^T \\ \vdots & \ddots & \vdots \\ F_nF_1^T & \cdots & F_nF_n^T \end{bmatrix}, \ \Sigma_i = \begin{bmatrix} F_iF_1^T & \cdots & F_iF_n^T \end{bmatrix}.$$

With the definition of ∞-norm, we have

$$\|\Sigma\|_\infty = \max_{1\leqslant i\leqslant n} \|\Sigma_i\|_\infty.$$

If node i knows $\|\Sigma_i\|_\infty$ at start, then after D_g-th iteration, each node can obtain

$$\max_{1\leqslant i\leqslant n} \|\Sigma_i\|_\infty.$$

Motivated by this observation, Algorithm 4.1 is proposed for the distributed computation of λ in (4.26).

Algorithm 4.1. *Distributed computation of* λ

```
Step 1   Communicate $F_j$ to node $i$ for $j \in \mathcal{N}_i$ and compute $F_iF_1^T$, if $j \notin \mathcal{N}_i$
Step 2   Calculate $Q_i = \| \begin{bmatrix} F_iF_1^T & \cdots & F_iF_n^T \end{bmatrix} \|_\infty$ at node $i$
Step 3   Communicate $Q_j$ to node $i$ for $j \in \mathcal{N}_i$
Step 4   Update $Q_i = \max\{Q_i, Q_{\mathcal{N}_i}\}$ at node $i$ with $Q_{\mathcal{N}_i} = \{Q_j | j \in \mathcal{N}_i\}$
Step 5   Repeat Step 3 and Step 4 $D_g$ times
Step 6   Determine $\lambda = \lambda_i = \frac{\alpha}{Q_i}$ at node $i$ with $0 < \alpha < 2$
```

With the value of λ, the distributed realization of (4.26) is formulated as

$$Z_i(\zeta+1) = Z_i(\zeta) + \lambda(\Lambda_i - \sum_{j\in\mathcal{S}_i} \Sigma_{ij} Z_j(\zeta)) \tag{4.34}$$

where $\mathcal{S}_i = \{i\} \bigcup \mathcal{N}_i$. It is clear that in (4.34), node i only uses its local and neighbors' information. Moreover, according to relation (4.5), stacking all (4.34) together, we obtain exactly the same equation as (4.26). Thus, (4.34) can be used as a distributed realization of (4.26). For the training phase, the distributed iterative computation (4.34) can be executed and terminate when

$$\|Z_i(\zeta+1) - Z_i(\zeta)\|_2 \leqslant \epsilon, \tag{4.35}$$

where ϵ is a predefined tolerance, is fulfilled.

Define new notations

$$I_4 = C_l^T \Sigma^{-1} = \begin{bmatrix} I_{4,1} & \cdots & I_{4,n} \end{bmatrix},\ I_5 = F_l^T \Sigma^{-1} = \begin{bmatrix} I_{5,1} & \cdots & I_{5,n} \end{bmatrix}. \tag{4.36}$$

After executing (4.34) until (4.35) is fulfilled, Z_i and $Z_i^T \Phi_i$ are obtained at node i, where Z_i can be $I_{4,i}$ or $I_{5,i}$ and Φ_i can be C_i or F_i. Notice that, for special case when Σ is block diagonal matrix with

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_n \end{bmatrix},$$

parameters $I_{4,i} = C_i^T \Sigma_i^{-1}$ and $I_{5,i} = F_i^T \Sigma_i^{-1}$ can be directly obtain at node i without performing distributed iterative computation (4.34).

4.2.1.2 Second Phase

As a result of the distributed iterative computation, $Z_i^T \Phi_i$ is obtained at node i, notice that our aim is to compute

$$Z^T \Phi = \begin{bmatrix} Z_1^T & \cdots & Z_n^T \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_n \end{bmatrix} = \sum_{j=1}^{n} Z_j^T \Phi_j \tag{4.37}$$

at each node. To this end, recall the distributed average consensus algorithm introduced in Chapter 2 by performing

$$\vartheta_i(\xi + 1) = w_{ii} \vartheta_i(\xi) + \sum_{j \in \mathcal{N}_i} w_{ij} \vartheta_j(\xi), \tag{4.38}$$

where ξ is the iteration number and matrix

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \tag{4.39}$$

is selected such that (2.21) and Lemma 2.5 are fulfilled. In this chapter, we apply the Metropolis-Hastings weights method (2.24) to compute W. From the computation (2.24), it is evident that the Metropolis-Hastings weights method only requires local and neighbours' information and thus can be realized in a distributed manner. Further, we have

$$\lim_{\xi \to \infty} W^{\xi} = \frac{1}{n} \mathbf{1}\mathbf{1}^T, \ \lim_{\xi \to \infty} \begin{bmatrix} \vartheta_1(\xi) \\ \vdots \\ \vartheta_n(\xi) \end{bmatrix} = \mathbf{1} \otimes (\frac{1}{n} \sum_{i=1}^{n} \vartheta_i(0)), \tag{4.40}$$

where $\mathbf{1} \in \mathcal{R}^n$ is a vector with all elements 1. It is clear that after running the distributed average consensus algorithm with initial value $\vartheta_i(0) = Z_i^T \Phi_i$ at node i, each node can obtain

$$n \left(\lim_{\xi \to \infty} \vartheta_i(\xi) \right) = \sum_{i=1}^{n} \vartheta_i(0) = \sum_{j=1}^{n} Z_j^T \Phi_j, \tag{4.41}$$

which is the aim in (4.37). After executing the distributed iterative computation and distributed average consensus, we obtain and save values of I_1, I_2 and I_3 at each node.

4.2.1.3 Third Phase

With I_1, I_2 and I_3 at hand, we calculate (4.13) in a distributed manner by the proposed iteration (4.23),

$$\begin{cases} X(\eta+1) = & \Xi(\eta) - \Psi(\eta) - \Psi^T(\eta) + \Pi(\eta) \\ \Omega(\eta) = & (X^{-1}(\eta) + I_1) \\ \Xi(\eta) = & A\Omega^{-1}(\eta)A^T \\ \Psi(\eta) = & A\Omega^{-1}(\eta)I_2^T E_l^T \\ \Pi(\eta) = & E_l(I - I_3 + I_2\Omega^{-1}(\eta)I_2^T)E_l^T \end{cases}.$$

After executing (4.23), each node obtains

$$\lim_{\eta\to\infty} X(\eta) = X, \ \lim_{\eta\to\infty} \Omega(\eta) = \Omega = (X^{-1} + I_1).$$

Finally, as a result of the offline training throughout the three phases, node i obtains

$$I_1, \ I_2, \ I_3, \ I_{4,i}, \ I_{5,i}, \ X, \ \Omega \tag{4.42}$$

in (4.15), (4.23) and (4.36).

4.2.2 Distributed Online Fault Detection

To realize online FD in a distributed manner, we apply (4.16) to reformulate (4.12) as

$$\begin{aligned} L &= (AXC_l^T + E_l F_l^T)R^{-1} = (AXC_l^T + E_l F_l^T)(\Sigma^{-1} - \Sigma^{-1}C_l\Omega^{-1}C_l^T\Sigma^{-1}) \\ &= (A - E_l I_2)\Omega^{-1}I_4 + E_l I_5. \end{aligned} \tag{4.43}$$

Define notation

$$L_i = (A - E_l I_2)\Omega^{-1}I_{4,i} + E_l I_{5,i}, \tag{4.44}$$

it is clear that after distributed offline training, node i can obtain L_i in (4.44) and

$$L = \begin{bmatrix} L_1 & \cdots & L_n \end{bmatrix}. \tag{4.45}$$

Also, define

$$\bar{r}_i(k) = y_i(k) - C_i\hat{x}_i(k), \ \bar{r}(k) = \begin{bmatrix} \bar{r}_1(k) \\ \vdots \\ \bar{r}_n(k) \end{bmatrix} = \begin{bmatrix} y_1(k) - C_1\hat{x}_1(k) \\ \vdots \\ y_n(k) - C_n\hat{x}_n(k) \end{bmatrix}. \tag{4.46}$$

For purpose of detection, node i runs (4.8) to have estimation $\hat{x}_i$. Further, combine (4.44) and (4.45) to modify state observer (4.8) in node i as

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + L\bar{r}(k) = A\hat{x}_i(k) + \sum_{i=1}^{n} L_i\bar{r}_i(k). \tag{4.47}$$

Given $\hat{x}_i(k)$ in time instant k, value $L_i\bar{r}_i(k)$ is obtain at node i, then

$$\sum_{i=1}^{n} L_i\bar{r}_i(k)$$

is identified by applying distributed average consensus (4.38) with $\vartheta_i = L_i\bar{r}_i(k)$, finally, $\hat{x}_i(k+1)$ can be obtained. Since all observers are identical, it holds that with the same initial value $\hat{x}_q(0) = x_0$ for $q = 1, \cdots, n$,

$$\hat{x}_i(k) = \hat{x}_j(k) = \hat{x}(k),\ i, j = 1, \cdots, n.$$

Thus, (4.47) is a distributed realization of (4.8). After the estimation of $x_i(k)$, $\bar{r}_i(k)$ is also obtained and used for computing term $r^T(k)r(k)$, which forms the evaluation function $J_i(k)$ in (4.14). With (4.16), it holds that

$$j_i(k) = r_i^T(k)r_i(k) = \bar{r}_i^T(k)R^{-1}\bar{r}_i(k) = \bar{r}_i^T(k)(\Sigma^{-1} - \Sigma^{-1}C_l\Omega^{-1}C_l^T\Sigma^{-1})\bar{r}(k). \tag{4.48}$$

Set notations

$$I_{6,i} = \bar{r}_i^T\Sigma^{-1},\ I_6 = \bar{r}^T\Sigma^{-1} = \begin{bmatrix} I_{6,1} & \cdots & I_{6,n} \end{bmatrix} \tag{4.49}$$

and transfer (4.48) into

$$j_i(k) = \sum_{i=1}^{n} I_{6,i}\bar{r}_i(k) - \left(\sum_{i=1}^{n} I_{4,i}\bar{r}_i(k)\right)^T \Omega^{-1} \left(\sum_{i=1}^{n} I_{4,i}\bar{r}_i(k)\right). \tag{4.50}$$

It is clear that in (4.49), I_6 has structure $\Lambda^T\Sigma^{-1}$ and $I_{6,i}$ can be obtained in node i by applying (4.34) online with $\Lambda_i = \bar{r}_i$. For (4.50), node i has $I_{6,i}\bar{r}_i(k)$ and $I_{4,i}\bar{r}_i(k)$, values

$$\sum_{i=1}^{n} I_{6,i}\bar{r}_i(k),\ \sum_{i=1}^{n} I_{4,i}\bar{r}_i(k)$$

are obtained by applying distributed average consensus (4.38) with $\vartheta_i = I_{6,i}\bar{r}_i(k)$ and $\vartheta_i = I_{4,i}\bar{r}_i(k)$, respectively. With $j_i(k)$ in hand, evaluation function

$$J_i(k) = \left(\frac{1}{m}\sum_{l=0}^{m-1} j_i(k+l)\right)^{1/2}, \tag{4.51}$$

is obtained, and it is clear that the combination of (4.50) and (4.51) is a distributed realization for $J(k)$ in (4.14).

Finally, we summarize the overall algorithm into Algorithm 4.2.

Algorithm 4.2. *Distributed FD in large-scale systems based on distributed average consensus*

```
Distributed Offline Training
For node $i$ with $i = 1, \cdots, n$
Step 1  Save $A_l$, $E_l$, $C_i$, $F_i$, $J_{th}$ and $D_g$
Step 2  Calculate $\lambda$ using Algorithm 4.1
Step 3  Calculate $W$ using Algorithm (2.24)
Step 4  Calculate $I_{4,i}$ and $I_{5,i}$ using distributed iterative computation (4.34)
Step 5  Calculate $I_1$, $I_2$ and $I_3$ using distributed average consensus (4.38)
Step 6  Calculate $X$ using (4.23)
Step 7  Calculate $\Omega$ in (4.15) and $L_i$ in (4.44)
Distributed Online FD
At time instant $k$, for node $i$ with $i = 1, \cdots, n$
Step 1  Measure $y_i(k)$ and compute $\bar{r}_i(k)$
Step 2  Calculate $I_{6,i}$ using distributed iterative computation (4.34)
Step 3  Calculate $\sum_{i=1}^{n} L_i\bar{r}_i(k)$, $\sum_{i=1}^{n} I_{4,i}\bar{r}_i(k)$ and $\sum_{i=1}^{n} I_{6,i}\bar{r}_i(k)$ using distributed
        average consensus (4.38)
Step 4  Calculate $J_i(k)$ using (4.50) and (4.51)
Step 5  Update $\hat{x}_i(k)$ using (4.47) for time instant $(k+1)$
Step 6  Make decision using (3.1)
```

4.3 Convergence Issue

Considering that the consensus should be achieved at each sampling interval during online computation in algorithm 4.2 through online implementation, the feasibility of implementing the proposed scheme in real-time could be questionable due to the convergence of average consensus.

One potential approach is to select the value of W in (4.39) in order to maximize the speed of convergence. Recall the objective of average consensus in (2.21),

$$x(k+1) = Wx(k), \ \lim_{k\to\infty} x(k) = \lim_{k\to\infty} W^k x(0) = \frac{1}{n}\mathbf{1}\mathbf{1}^\top x(0).$$

Set

$$x^* = \frac{1}{n}\mathbf{1}\mathbf{1}^\top x(0)$$

and use per-step convergence factor

$$r_{step}(W) = \sup_{x(k)\neq x^*} \frac{\|x(k+1) - x^*\|_2}{\|x(k) - x^*\|_2}$$

as index to measure the speed of convergence. To achieve the fastest convergence, it is imperative to minimize $r_{step}(W)$, a goal addressed through the subsequent norm minimization problem:

$$\begin{aligned} \min \quad & \|W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\|_2 \\ \text{s.t.} \quad & W \in \mathcal{S},\ W\mathbf{1} = \mathbf{1},\ \mathbf{1}^\top W = \mathbf{1}^\top \end{aligned}$$

where set $\mathcal{S}$ is defined in (2.20). Since this problem is convex, it can be solved efficiently and globally [112].

The other viable approach involves reducing the amount of information used for detection in node i. This approach presents a trade-off between expediting decision-making with limited information and waiting for extended computation times based on more information to enhance decision accuracy. Opting for less information speeds up the convergence, but this also impacts detection performance. In detection tasks, the goal is to amplify the impact of faults on the residual while minimizing the influence of disturbances on the residual signal. From this observation,

$$J(y^*) = \max_{L,V} \frac{\|G_{rf}\|_\infty}{\|G_{rd}\|_\infty}, \tag{4.52}$$

where L and V are solved by (3.57) for

$$\begin{aligned} & x(k+1) = Ax(k) + E_l d_l(k) + E_f f(k),\ y^*(k) = C^* x(k) + F^* d_l(k), \\ & G_{rd}(z) = (A - LC^*, E_l - LF^*, VC^*, VF^*)\,,\ G_{rf}(z) = (A - LC^*, E_f, VC^*, 0) \end{aligned}$$

with the selected information y^* and corresponding C^* and F^* compared with (4.7), can be used as performance index to measure the detection performance. And the problem of information selection can be formulated as

$$\text{Select the lest information } y^*, \text{ s.t. } J(y^*) \geqslant J^*,$$

where J^* is a pre-defined lower bound for detection performance. In Chapter 5, we will employ a comparable performance index compared with (4.52) for optimization purpose, while in Chapter 6, a comprehensive exploration of the concept of information selection will be discussed.

4.4 Example

Consider a dynamic model (4.1) with $x(k) \in \mathcal{R}^{10}$, $d_p \in \mathcal{R}^{10}$, and $d_m \in \mathcal{R}^{10}$. In the applied sensor network, we have $n = 10$ sensors and $y_i \in \mathcal{R}^1$ for $i = 1, \cdots, 10$. The corresponding matrices are

$$C_1 = \begin{bmatrix} 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ C_2 = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} 0 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ C_4 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$C_5 = \begin{bmatrix} 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},\ C_6 = \begin{bmatrix} 1 & 0 & 1 & 0 & 3 & 1 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$C_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 5 & 0 & 0 \end{bmatrix},\ C_8 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 6 & 1 & 1 & 0 \end{bmatrix},$$

$$C_9 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & 1 & 0 \end{bmatrix},\ C_{10} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$F_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$F_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix},$$

$$F_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix},$$

$$F_9 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$

$$F_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$A = \begin{bmatrix} 0.2 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.1 & 0.0 \\ 0.3 & 0.1 & 0.1 & 0.2 & 0.3 & 0.0 & 0.2 & 0.0 & 0.2 & -0.1 \\ 0.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.0 & 0.0 & 0.0 & 0.2 \\ 0.1 & 0.1 & 0.0 & 0.2 & 0.0 & 0.1 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.2 & 0.0 & 0.1 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.1 & 0.2 & 0.0 & 0.2 & 0.0 & 0.2 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.1 & 0.1 & 0.2 & 0.0 & 0.1 & 0.0 & 0.1 \\ 0.1 & 0.0 & 0.0 & 0.1 & 0.1 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.0 & 0.0 & 0.0 & 0.1 & 0.2 & 0.0 & 0.3 & 0.1 & 0.0 \\ 0.2 & 0.0 & 0.1 & 0.1 & 0.0 & 0.1 & 0.2 & 0.1 & 0.1 & 0.3 \end{bmatrix},\ E_f = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$E_l = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

According to matrices F_i with $i = 1, \ldots, 10$ and (4.5), the communication topology of the sensor network is shown in Figure 4.1. Moreover, the diameter of the graph in Figure 4.1 is $D_g = 3$, and the corresponding Laplacian matrix is shown in (4.4). The observer gain L in (4.12) is

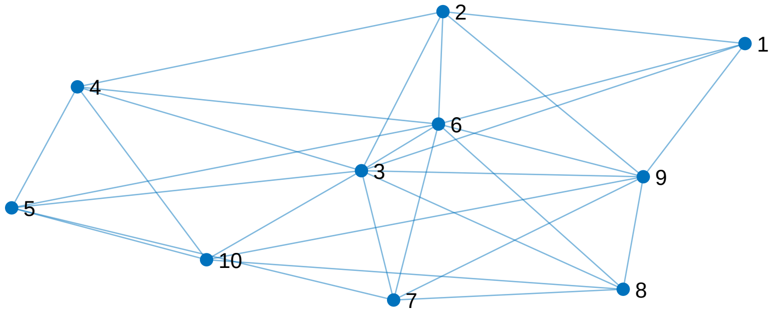


Figure 4.1: Communication topology of sensor network

$$\begin{bmatrix} 0.112 & -0.063 & -0.006 & 0.047 & -0.032 & -0.001 & 0.029 & -0.001 & -0.011 & 0.046 \\ 0.052 & -0.029 & 0.003 & 0.020 & 0.100 & 0.012 & -0.041 & 0.040 & 0.057 & -0.007 \\ -0.034 & 0.017 & 0.022 & -0.036 & 0.016 & 0.053 & 0.040 & -0.023 & -0.038 & 0.101 \\ -0.011 & 0.010 & -0.019 & 0.004 & 0.020 & 0.027 & 0.057 & 0.005 & -0.021 & 0.106 \\ -0.035 & 0.040 & -0.009 & -0.030 & 0.049 & -0.002 & 0.034 & 0.005 & -0.008 & 0.027 \\ -0.059 & 0.023 & -0.005 & 0.007 & 0.011 & 0.053 & 0.001 & 0.020 & 0.012 & 0.098 \\ -0.046 & 0.037 & -0.024 & -0.031 & 0.071 & 0.025 & 0.038 & -0.012 & -0.006 & 0.033 \\ -0.018 & 0.011 & -0.032 & 0.023 & -0.016 & 0.047 & 0.038 & 0.006 & -0.007 & 0.111 \\ 0.007 & 0.016 & -0.056 & 0.018 & 0.051 & 0.002 & 0.055 & 0.001 & 0.018 & 0.019 \\ -0.022 & 0.034 & -0.019 & -0.014 & -0.012 & 0.048 & 0.070 & -0.002 & -0.045 & 0.163 \end{bmatrix}$$

Apply Algorithm 4.1 and select $\alpha = 1$ to obtain $\lambda = 0.0556$. Apply the Metropolis-Hastings weights method (2.24) to compute W matrix for distributed average consensus. The result is

$$W = \begin{bmatrix} 0.497 & 0.167 & 0.100 & 0.000 & 0.000 & 0.111 & 0.000 & 0.000 & 0.125 & 0.000 \\ 0.167 & 0.331 & 0.100 & 0.167 & 0.000 & 0.111 & 0.000 & 0.000 & 0.125 & 0.000 \\ 0.100 & 0.100 & 0.100 & 0.100 & 0.100 & 0.100 & 0.100 & 0.100 & 0.100 & 0.100 \\ 0.000 & 0.167 & 0.100 & 0.289 & 0.167 & 0.111 & 0.000 & 0.000 & 0.000 & 0.167 \\ 0.000 & 0.000 & 0.100 & 0.167 & 0.289 & 0.111 & 0.167 & 0.000 & 0.000 & 0.167 \\ 0.111 & 0.111 & 0.100 & 0.111 & 0.111 & 0.122 & 0.111 & 0.111 & 0.111 & 0.000 \\ 0.000 & 0.000 & 0.100 & 0.000 & 0.167 & 0.111 & 0.331 & 0.167 & 0.125 & 0.000 \\ 0.000 & 0.000 & 0.100 & 0.000 & 0.000 & 0.111 & 0.167 & 0.331 & 0.125 & 0.167 \\ 0.125 & 0.125 & 0.100 & 0.000 & 0.000 & 0.111 & 0.125 & 0.125 & 0.164 & 0.125 \\ 0.000 & 0.000 & 0.100 & 0.167 & 0.167 & 0.000 & 0.000 & 0.167 & 0.125 & 0.275 \end{bmatrix}$$

Given $\delta_d = 7.1929$, we set

$$J_i(k) = (\frac{1}{6}\sum_{l=0}^{5} j_i(k+l))^{1/2}, \;\; J_{th} = \sqrt{\frac{1}{6}\delta_d} = 2.9365,$$

sampling time $T = 0.02$s for online detection. With an additive step fault happening at 6s, simulation results of the evaluation function of the centralized solution J and the evaluation functions for sensor nodes 1−10 are shown as J_1−J_{10} in Figure 4.2−4.6, respectively.

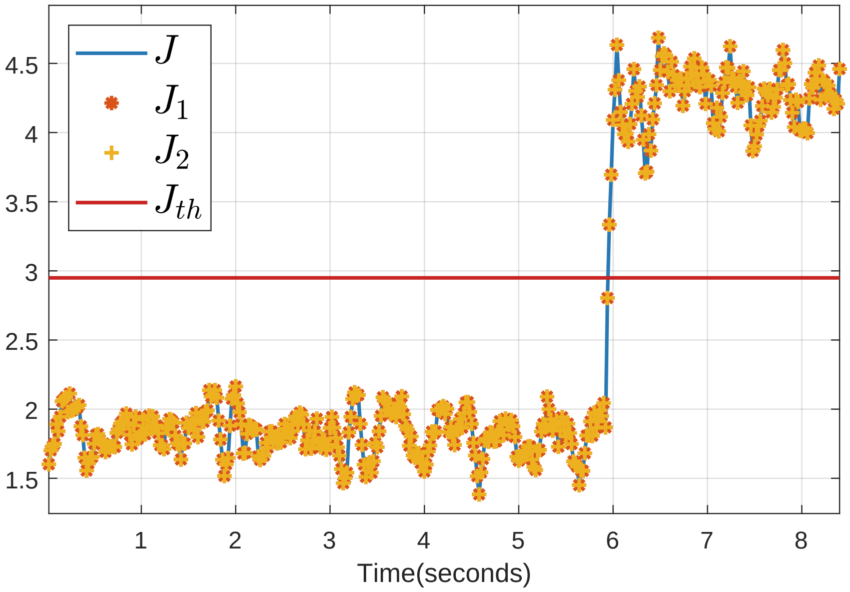


Figure 4.2: Simulation results for nodes 1 and 2

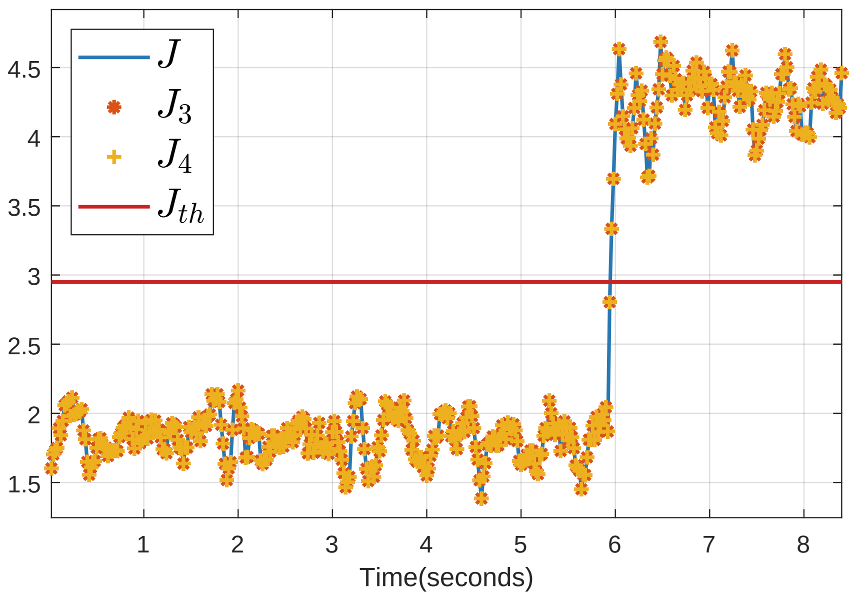


Figure 4.3: Simulation results for nodes 3 and 4

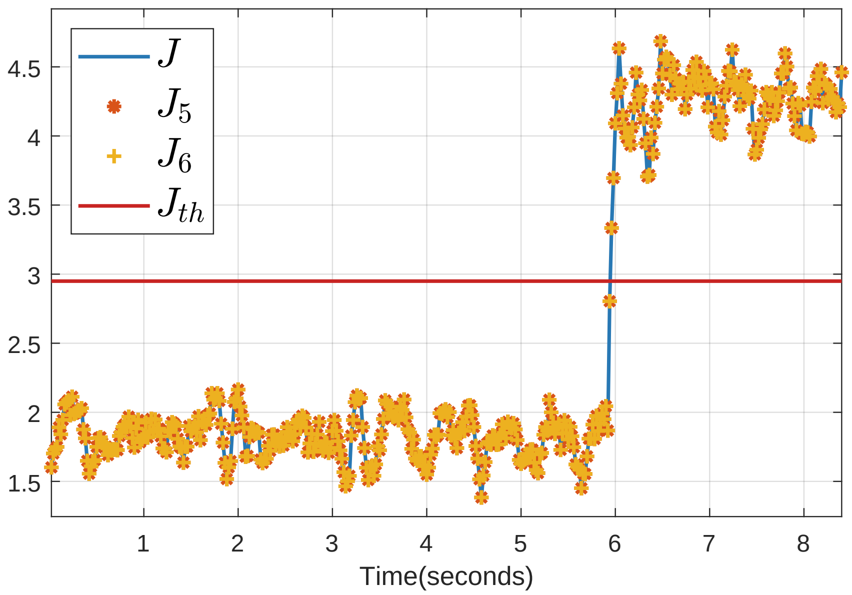


Figure 4.4: Simulation results for nodes 5 and 6

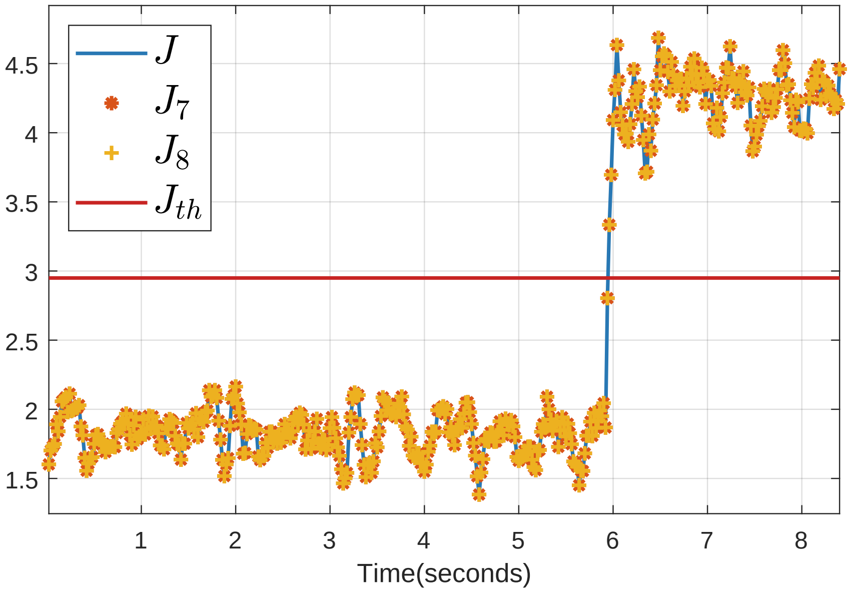


Figure 4.5: Simulation results for nodes 7 and 8

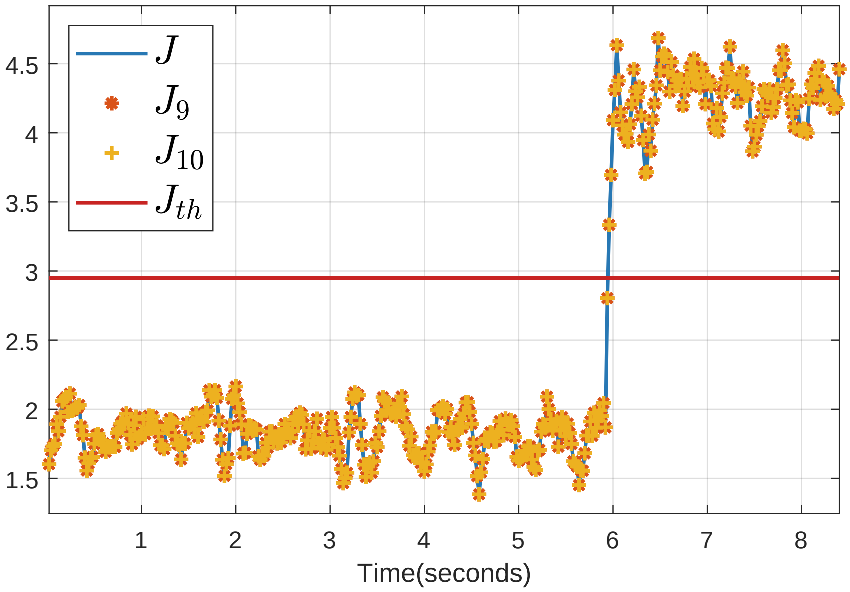


Figure 4.6: Simulation results for nodes 9 and 10

The figures indicate that our proposed distributed FD scheme can achieve a similar result when compared with the centralized approach, which is plotted as J in each figure. However, with the limited online execution time, both distributed iterative computation and distributed average consensus may not converge to the accuracy value, which may lead to inaccuracy for the online implementation as shown in Figure 4.7, which is an enlarged local figure of Figure. 4.2.

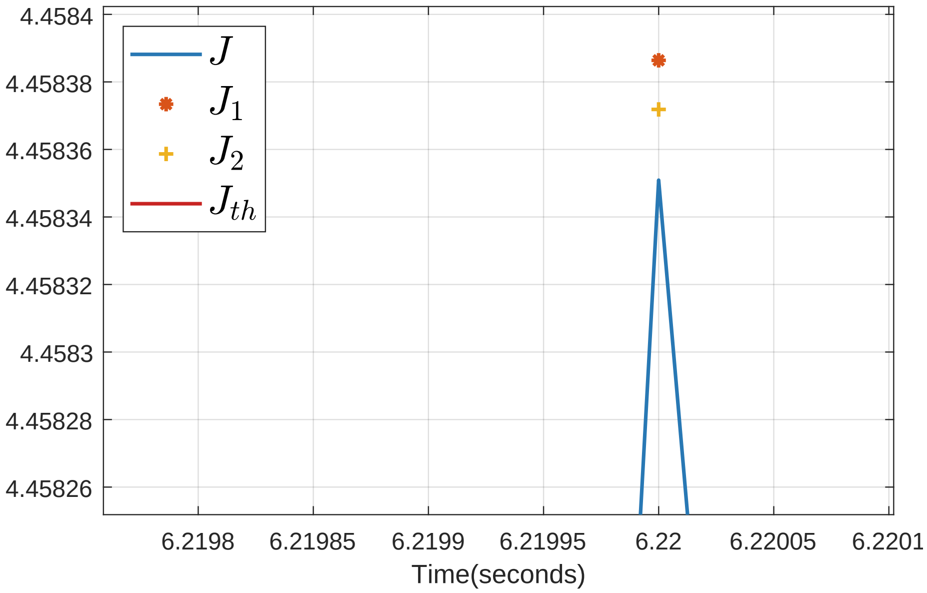


Figure 4.7: Comparison of evaluation functions

4.5 Concluding Remarks

This chapter refers to a novel distributed FD scheme proposed for large-scale systems that are affected by deterministic disturbances using sensor networks. The original model (4.1) ignores the input signal since it is assumed in this case all sensors know the input signal. Theoretical analysis, which is based on centralized optimal FD scheme, distributed iterative computation and distributed average consensus, and the simulation results show that the proposed scheme can detect fault effectively when compared with the centralized approach. We would like to remark that, offline training and online implementation are all realized in a distributed way, which means that the sensor node can only use its local and neighbors' information for data fusion and corresponding purpose. The pioneering work of the distributed realization of optimal FD for stochastic processes is shown in [27], and the related distributed Kalman filtering is proposed in [86].

5 Distributed Fault Detection in Interconnected Systems Based on Linear Matrix Inequality

The distributed realization of optimal FD proposed in Chapter 4 is based on distributed average consensus. During online implementation, the distributed computation of the evaluation function needs time to converge since it is realized via distributed average consensus and requires the information among the whole network. Such properties may lead to a longer mean time to detection (MTTD). In this chapter, a distributed state observer is applied for distributed FD. It has a simpler structure and online implementation form when compared with the method based on average consensus. We first introduce the design issue of observer parameters. Unlike the complicated methods in [94] to directly meet restrictive requirements. LMI-based numerical methods are applied to guarantee the convergence condition. Then, a post-filter is designed to satisfy our performance requirement, that is, suppress the influence of deterministic disturbances on residual signal and meanwhile enhance its sensibility to faults.

5.1 Distributed State Observer Design

The large-scale interconnected system under consideration consists of M subsystems with the same sampling time T. Each subsystem G_i, where $i = 1, \cdots, M$, is modelled by

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_i u_i(k) + \sum_{j\in\mathcal{N}_i} A_{ij}x_j(k) + E_{di}d_i(k) + E_{fi}f_i(k), \\ y_i(k) &= C_i x_i(k) + F_{di}d_i(k) + F_{fi}f_i(k). \end{aligned} \tag{5.1}$$

where $x_i(k) \in \mathcal{R}^{n_i}$ is the local state, $u_i(k) \in \mathcal{R}^{q_i}$ is the local input, $y_i(k) \in \mathcal{R}^{m_i}$ is the local measurement, $d_i(k) \in \mathcal{R}^{n_i}$ denotes the disturbance.

For the purpose of FD, each subsystem is locally implemented with a distributed state observer [9, 94] to estimate the local states using its local and neighbours' information.

The distributed state observer has the following structure

$$\begin{aligned} \hat{x}_i(k+1) &= A_{ii}\hat{x}_i(k) + B_i u_i(k) + L_{ii}(y_i(k) - C_i\hat{x}_i(k)) + S_{i,j}(k) \\ S_{i,j}(k) &= \sum_{j\in\mathcal{N}_i} (A_{ij}\hat{x}_j(k) + L_{ij}(y_j(k) - C_j\hat{x}_j(k))) \\ \hat{y}_i(k) &= C_i\hat{x}_i(k) \end{aligned} . \tag{5.2}$$

From (5.2), the distributed state observer for the subsystem i depends only on local measurement y_i and its neighbors' information $\hat{x}_j$, y_j wtih $j \in \mathcal{N}_i$. For further analysis, define the local state estimation error as

$$e_i(k) = x_i(k) - \hat{x}_i(k).$$

Combine (5.1) and (5.2), it holds that

$$\begin{aligned} e_i(k+1) &= (A_{ii} - L_{ii}C_i)e_i(k) + (E_{di} - L_{ii}F_{di})d_i(k) + (E_{fi} - L_{ii}F_{fi})f_i(k) + \bar{S}_{i,j}(k), \\ \bar{S}_{i,j}(k) &= \sum_{j\in\mathcal{N}_i} ((A_{ij} - L_{ij}C_j)e_j(k) - L_{ij}F_{dj}d_j(k) - L_{ij}F_{fj}f_j(k)) . \end{aligned} \tag{5.3}$$

Concatenate all e_i together to model the overall error dynamic as

$$e(k+1) = (A - LC)e(k) + (E_d - LF_d)d(k) + (E_f - LF_f)f(k), \tag{5.4}$$

where

$$\begin{gathered} e(k) = \begin{bmatrix} e_1(k) \\ \vdots \\ e_M(k) \end{bmatrix}, d(k) = \begin{bmatrix} d_1(k) \\ \vdots \\ d_M(k) \end{bmatrix}, f(k) = \begin{bmatrix} f_1(k) \\ \vdots \\ f_M(k) \end{bmatrix}, \\ A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{21} & A_{22} & \cdots & A_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MM} \end{bmatrix}, L = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1M} \\ L_{21} & L_{22} & \cdots & L_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ L_{M1} & L_{M2} & \cdots & L_{MM} \end{bmatrix}, \\ C = \mathrm{diag}(C_1, \ldots, C_M), E_d = \mathrm{diag}(E_{d1}, \cdots, E_{dM}), F_d = \mathrm{diag}(F_{d1}, \cdots, F_{dM}), \\ E_f = \mathrm{diag}(E_{f1}, \cdots, E_{fM}), F_f = \mathrm{diag}(F_{f1}, \cdots, F_{fM}). \end{gathered} \tag{5.5}$$

Notation diag() stands for a block diagonal matrix with

$$\mathrm{diag}(M_1, \ldots, M_M) = \begin{bmatrix} M_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & M_M \end{bmatrix}$$

Define set

$$\mathcal{S}_j = \mathcal{N}_j \bigcup \{j\}. \tag{5.6}$$

From (5.2) and (5.3), matrices $A_{ij} \neq 0$ and $L_{ij} \neq 0$ when $j \in \mathcal{S}_i$ and it also means that G_i and G_j are physically connected. Otherwise, $A_{ij} = 0$ and $L_{ij} = 0$ when $j \notin \mathcal{S}_i$ and they are disconnected. In nominal case, error dynamic (5.4) reduces to

$$e(k+1) = (A - LC)e(k). \tag{5.7}$$

In order to guarantee condition (3.31), matrix L should be designed such that $A - LC$ is Schur matrix. Different from the centralized solution of L, the additional restriction

$$L_{ij} = 0 \text{ if } j \notin \mathcal{S}_i$$

makes the design procedure a complex and challenging task. In the following, we introduce LMI conditions for designing L to guarantee condition (3.31).

Theorem 5.1. *Given system* (5.1) *and distributed observer* (5.2)*, its nominal and global error dynamic* (5.7) *guarantees the condition* (3.31) *if there exist symmetric positive definite matrices $P > 0$, $Q > 0$ and a matrix X, where $P = \mathrm{diag}(P_1, P_2, \ldots, P_M)$, $Q = \mathrm{diag}(Q_1, Q_2, \ldots, Q_M)$, $P_i \in \mathcal{R}^{n_i \times n_i}$, $X_{ij} = 0$ if $j \notin \mathcal{S}_i$, $Q_i \in \mathcal{R}^{n_i \times n_i}$ and $X_{ij} \in \mathcal{R}^{n_i \times p_j}$, satisfy conditions*

$$\begin{bmatrix} -P & * \\ QA - XC & P - 2Q \end{bmatrix} < 0,$$

$$QA - XC = \begin{bmatrix} Q_1A_{11} - X_{11}C_1 & Q_1A_{12} - X_{12}C_2 & \cdots & Q_1A_{1M} - X_{1M}C_M \\ Q_2A_{21} - X_{21}C_1 & Q_2A_{22} - X_{22}C_2 & \cdots & Q_2A_{2M} - X_{2M}C_M \\ \vdots & \vdots & \ddots & \vdots \\ Q_MA_{M1} - X_{M1}C_1 & Q_MA_{M2} - X_{M2}C_2 & \cdots & Q_MA_{MM} - X_{MM}C_M \end{bmatrix}.$$

And $L_{ij} = Q_i^{-1}X_{ij}$.

Proof. From Lemma 2.7, if there exist positive definite matrices $P = \mathrm{diag}(P_1, P_2, \ldots, P_M)$ and $Q = \mathrm{diag}(Q_1, Q_2, \ldots, Q_M)$ such that

$$\begin{bmatrix} -P & * \\ Q(A - LC) & P - 2Q \end{bmatrix} < 0, \tag{5.8}$$

matrix $A - LC$ is Schur matrix. Set $X = QL$, nonlinear inequality (5.8) is transferred into a linear one. Due to the diagonal structure of Q, it is straightforward that $X_{ij} = Q_iL_{ij}$ and $L_{ij} = Q_i^{-1}X_{ij}$. □

Since the method provided in Theorem 5.1 is only sufficient condition and conservative, in the following, we will give the another method to calculate the observer gain matrices. This method is inspired by an inequality which is expressed by the following lemma.

Lemma 5.1. *[9] A matrix M is composed of block matrices $M_{ij} \in \mathbb{R}^{n_i x n_j}$. If $j \notin \mathcal{N}_i$, then $M_{ij} = 0$. For $i = 1, 2, \ldots, M$, consider block-diagonal matrix $H = diag(H_1, H_2, \ldots, H_M)$, where $H_i \geqslant 0$, and block-diagonal matrix $H^+ = diag(H_1^+, H_2^+$ with elements*

$$H_i^+ = \sum_{j \in \mathcal{S}_i} (\sqrt{c_j} M_{ij}) H_j (\sqrt{c_j} M_{ij})^T, \tag{5.9}$$

where $c_j = |\mathcal{S}_j|$. It holds that

$$H^+ \geq MHM^T \tag{5.10}$$

Now, we apply Lemma 5.1 to the second design method.

Theorem 5.2. *Given system (5.1), $c_j = |\mathcal{S}_j|$, and distributed observer (5.2), its nominal and global error dynamic (5.7) guarantees the condition (3.31) if there exist $X_i > 0$ for $i = 1, 2, \cdots, M$ and Q_{ij} for $j \in \mathcal{S}_i$ such that*

$$\begin{bmatrix} -X_i & \sqrt{c_i}(X_i A_{ii} - Q_{ii} C_i) & \cdots & \sqrt{c_{i_{n_i}}}(X_i A_{ii_{n_i}} - Q_{ii_{n_i}} C_{i_{n_i}}) \\ * & -X_i & 0 & 0 \\ * & * & \ddots & 0 \\ * & * & * & -X_{i_{n_i}} \end{bmatrix} < 0,$$

where set $\mathcal{N}_i$ is ordered as $\mathcal{N}_i = \{i_1, \cdots, i_{n_i}\}$. And $L_{ij} = (X_i)^{-1} Q_{ij}$.

Proof. From Lemma 2.7, if there exist positive definite matrices $P = \text{diag}(P_1, P_2, \ldots, P_M)$ such that

$$(A - LC)P(A - LC)^T - P < 0, \tag{5.11}$$

matrix $A - LC$ is Schur matrix. Set $M_{ij} = A_{ij} - L_{ij} C_j$ and from Lemma 5.1, if

$$\sum_{j \in \mathcal{S}_i} (\sqrt{c_j} M_{ij}) P_j (\sqrt{c_j} M_{ij})^T - P_i < 0, \; i = \{1, \cdots, M\}, \tag{5.12}$$

inequality (5.11) holds. From (5.12) and Lemma 2.6, we have

$$\begin{bmatrix} -P_i & \sqrt{c_i} M_{ii} & \sqrt{c_{i_1}} M_{ii_1} & \cdots & \sqrt{c_{i_{n_i}}} M_{ii_{n_i}} \\ * & -P_i^{-1} & 0 & \cdots & 0 \\ * & * & -P_{i_1}^{-1} & \ddots & \vdots \\ * & * & * & \ddots & 0 \\ * & * & * & * & -P_{i_{n_i}}^{-1} \end{bmatrix} < 0, \tag{5.13}$$

Set $X_i = P_i^{-1}$ and transfer matrix

$$T = \begin{bmatrix} X_i & 0 \\ 0 & I \end{bmatrix}$$

We do pre-multiplication and post-multiplication for both sides of (5.13) by T to obtain

$$\begin{bmatrix} -X_i & \sqrt{c_i}X_iM_{ii} & \sqrt{c_{i_1}}X_iM_{ii_1} & \cdots & \sqrt{c_{in_i}}X_iM_{ii_{n_i}} \\ * & -X_i & 0 & \cdots & 0 \\ * & * & -X_{i_1} & \ddots & \vdots \\ * & * & * & \ddots & 0 \\ * & * & * & * & -X_{i_{n_i}} \end{bmatrix} < 0.$$

With $Q_{ij} = X_iL_{ij}$, the theorem is proved. □

Notice that the proposed two theorems are only sufficient LMI conditions to guarantee the asymptotic stability. And the influences of disturbances and faults are not considered in the computation of observer gain matrices. This motivates us to implement post-filter for the local residual signal

$$r_i(k) = y_i(k) - C_i\hat{x}_i(k). \tag{5.14}$$

In the next section, the post-filter is designed to achieve the robustness of the residual against disturbances and meanwhile increase its sensitivity to possible faults.

5.2 Post-Filter Design

In this section, we propose methods to design the post-filter. Recall (5.3) and (5.14), we have, for $i = 1, 2, \ldots, M$,

$$\begin{aligned} e_i(k+1) &= (A_{ii} - L_{ii}C_i)e_i(k) + (E_{di} - L_{ii}F_{di})d_i(k) + (E_{fi} - L_{ii}F_{fi})f_i(k) + \bar{S}_{i,j}(k), \\ \bar{S}_{i,j}(k) &= \sum_{j\in\mathcal{N}_i} (A_{ij} - L_{ij}C_j)e_j(k) - L_{ij}F_{dj}d_j(k) - L_{ij}F_{fj}f_j(k)) \,, \\ r_i(k) &= C_ie_i(k) + F_{di}d_i(k) + F_{fi}f_i(k). \end{aligned} \tag{5.15}$$

For node i, consider the influence of f_i and disturbance d. Besides the notations in (5.5), set

$$\tilde{A} = A - LC, \tilde{E}_d = E_d - LF_d, \tilde{E}_{fi} = \begin{bmatrix} (-L_{1i}F_{fi})^T & \cdots & (E_{fi} - L_{ii}F_{fi})^T & \cdots & (-L_{Mi}F_{fi})^T \end{bmatrix}^T,$$
$$\tilde{C}_i = \begin{bmatrix} 0 & \cdots & C_i & \cdots & 0 \end{bmatrix}, \tilde{F}_{d_i} = \begin{bmatrix} 0 & \cdots & F_{di} & \cdots & 0 \end{bmatrix},$$

and modify (5.15) into the global form of error dynamic with local residual as

$$e(k+1) = \tilde{A}e(k) + \tilde{E}_d d(k) + \tilde{E}_{fi} f_i(k), \tag{5.16}$$

$$r_i(k) = \tilde{C}_i e(k) + \tilde{F}_{di} d(k) + F_{fi} f_i(k). \tag{5.17}$$

In order to perform more accurate FD, it is desired that the residual signal is sensitive to faults, meanwhile robust against disturbances. For our purpose, use notations

$$r_{i,s}(k) = \begin{bmatrix} r_i(k-s) \\ r_i(k-s+1) \\ \vdots \\ r_i(k) \end{bmatrix}, d_s(k) = \begin{bmatrix} d(k-s) \\ d(k-s+1) \\ \vdots \\ d(k) \end{bmatrix}, f_{i,s}(k) = \begin{bmatrix} f_i(k-s) \\ f_i(k-s+1) \\ \vdots \\ f_i(k) \end{bmatrix}$$

and

$$H_{e,s} = \begin{bmatrix} \tilde{C}_i \\ \tilde{C}_i\tilde{A} \\ \vdots \\ \tilde{C}_i\tilde{A}^s \end{bmatrix}, H_{di,s} = \begin{bmatrix} \tilde{F}_{di} & 0 & \cdots & 0 \\ \tilde{C}_i\tilde{E}_d & \tilde{F}_{di} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \tilde{C}_i\tilde{A}^{s-1}\tilde{E}_d & \tilde{C}_i\tilde{A}^{s-2}\tilde{E}_d & \cdots & \tilde{F}_{di} \end{bmatrix},$$

$$H_{fi,s} = \begin{bmatrix} F_{fi} & 0 & \cdots & 0 \\ \tilde{C}_i\tilde{E}_{fi} & F_{fi} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \tilde{C}_i\tilde{A}^{s-1}\tilde{E}_{fi} & \tilde{C}_i\tilde{A}^{s-2}\tilde{E}_{fi} & \cdots & F_{fi} \end{bmatrix}$$

to modify the original model (5.16) and (5.17) as lifted model

$$r_{i,s}(k) = H_{e,s}e(k-s) + H_{di,s}d_s(k) + H_{fi,s}f_{i,s}(k) \tag{5.18}$$

Based on the equation (5.18), a post-filter $v_{i,s}$ is designed to cancel the influence of initial state $e(k-s)$, reduce the influence of the disturbance on residual, and increase the sensitivity of residual to faults. Denote Q_i as the base matrix of parity space with

$$Q_i H_{e,s} = 0.$$

In order to cancel the influence of initial state $e(k-s)$, it holds that

$$v_{i,s} = \tilde{v}_{i,s} Q_i,$$

where $\tilde{v}_{i,s}$ is a vector to be chosen later. Do pre-multiplication to both sides of (5.18) by $v_{i,s}$,

$$\tilde{v}_{i,s} Q_i r_{i,s}(k) = \tilde{v}_{i,s} Q_i H_{di,s} d_s(k) + \tilde{v}_{i,s} Q_i H_{fi,s} f_{i,s}(k).$$

Based on the above discussion, set the performance index for $v_{i,s}$ as

$$\max \frac{\text{influence of the fault}}{\text{influence of the disturbances}}$$

$$\Updownarrow$$

$$\mathcal{J} = \max_{\tilde{v}_{i,s}} \frac{\tilde{v}_{i,s} Q_i H_{fi,s} H_{fi,s}^T Q_i^T \tilde{v}_{i,s}^T}{\tilde{v}_{i,s} Q_i H_{di,s} H_{di,s}^T Q_i^T \tilde{v}_{i,s}^T}, v_{i,s} = \tilde{v}_{i,s} Q_i \tag{5.19}$$

We assume that

$$\text{rank}(Q_i H_{di,s}) = \text{row number of } (Q_i H_{di,s}). \tag{5.20}$$

Otherwise, vector $\tilde{v}_{i,s}$ can be chosen such that $\tilde{v}_{i,s} Q_i H_{di,s} = 0$ and we can directly decouple disturbance from residual.

Theorem 5.3. *Assume $(Q_i H_{di,s})$ has full column rank, do eigen decomposition*

$$Q_i H_{di,s} H_{di,s}^T Q_i^T = U \Sigma U^T.$$

Denote the largest eigenvalue and corresponding eigenvector of

$$\Sigma^{-\frac{1}{2}} U^T Q_i H_{fi,s} H_{fi,s}^T Q_i^T U \Sigma^{-\frac{1}{2}}$$

as $\bar{\lambda}$ and $\bar{v}_{i,s}$, we have

$$\tilde{v}_{i,s}^* = \bar{v}_{i,s} \Sigma^{-\frac{1}{2}} U^T, \mathcal{J}^* = \bar{\lambda} \tag{5.21}$$

as the optimal solution to (5.19). And the post-filter $v_{i,s}$ is

$$v_{i,s} = \tilde{v}_{i,s}^* Q_i.$$

Proof. See Theorem 2.5 in [54]. □

After the design of the post-filter $v_{i,s}$, we have

$$\bar{r}_{i,s}(k) = v_{i,s} r_{i,s}(k) = v_{i,s}(H_{e,s} e(k-s) + H_{di,s} d_s(k) + H_{fi,s} f_{i,s}(k)). \tag{5.22}$$

Since the generation of $\bar{r}_{i,s}(k)$ needs mean time $(s+1)$, we use DO for online realization of the proposed PSA. A one-to-one mapping between PSA and DO is introduced in the following Theorem.

Theorem 5.4. [24] *Given system model (5.16) and (5.17) and a parity vector*

$$v_{i,s} = \begin{bmatrix} v_{i,s,0} & v_{i,s,1} & \cdots & v_{i,s,s} \end{bmatrix}$$

the dead-beat DO

$$\begin{aligned} z_i(k+1) &= G_i z_i(k) + L_i r_i(k) \\ \epsilon_i(k) &= v_i r_i(k) - w_i z_i(k) \end{aligned} \tag{5.23}$$

$$G_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, \; L_i = -\begin{bmatrix} v_{i,s,0} \\ v_{i,s,1} \\ \vdots \\ v_{i,s,s-1} \end{bmatrix}, \; v_i = v_{i,s,s}, \; w_i = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}$$

provides a realization for

$$\bar{r}_{i,s}(k) = v_{i,s} r_{i,s}(k)$$

in (5.22).

5.3 Residual Evaluation and Threshold Setting

After applying the proposed methods to design distributed state estimator and post-filter, from (5.16), (5.17), and (5.23), we have

$$\begin{aligned} e(k+1) &= \tilde{A}e(k) + \tilde{E}_d d(k) + \tilde{E}_{fi} f_i(k) \\ r_i(k) &= \tilde{C}_i e(k) + \tilde{F}_{di} d(k) + F_{fi} f_i(k) \\ z_i(k+1) &= G_i z_i(k) + L_i r_i(k) \\ \epsilon_i(k) &= v_i r_i(k) - w_i z_i(k) \end{aligned} . \tag{5.24}$$

Notice that (5.24) stands for the transfer function from disturbances and faults to corresponding local residual based on designed distributed observer and post-filter and ϵ_i is finally utilised for FD. For threshold setting, modify (5.24) as

$$\begin{aligned} \begin{bmatrix} e(k+1) \\ z_i(k+1) \end{bmatrix} &= \begin{bmatrix} \tilde{A} & 0 \\ L_i\tilde{C}_i & G_i \end{bmatrix} \begin{bmatrix} e(k) \\ z_i(k) \end{bmatrix} + \begin{bmatrix} \tilde{E}_d \\ L_i\tilde{F}_{di} \end{bmatrix} d(k) + \begin{bmatrix} \tilde{E}_{fi} \\ L_i F_{fi} \end{bmatrix} f_i(k) \\ \epsilon_i(k) &= \begin{bmatrix} v_i\tilde{C}_i & -w_i \end{bmatrix} \begin{bmatrix} e(k) \\ z_i(k) \end{bmatrix} + v_i\tilde{F}_{di} d(k) + v_i F_{fi} f_i(k) \end{aligned} . \tag{5.25}$$

Based on (5.25) , compute the transfer function from d to ϵ_i as

$$G_{\epsilon_i d} = \left(\begin{bmatrix} \tilde{A} & 0 \\ L_i\tilde{C}_i & G_i \end{bmatrix}, \begin{bmatrix} \tilde{E}_d \\ L_i\tilde{F}_{di} \end{bmatrix}, \begin{bmatrix} v_i\tilde{C}_i & -w_i \end{bmatrix}, v_i\tilde{F}_{di} \right). \tag{5.26}$$

In fault-free case, it holds that

$$\|\epsilon_i(k)\|_2 \leqslant \|G_{\epsilon_i d}\|_\infty \|d(k)\|_2.$$

Based on (3.63), in practice, set

$$J_i = \|\epsilon_i(k)\|_{RMS} = \sqrt{\frac{1}{s}\sum_{j=k-s+1}^{k} \epsilon_i^T(j)\epsilon_i(j)},\ J_{th,i} = \gamma_i\|d(k)\|_{RMS},\ \|G_{\epsilon_i d}\|_\infty = \gamma_i \tag{5.27}$$

for FD. With given $\|d\|_2 \leq \delta_d$,

$$\|d(k)\|_{RMS} \leqslant \sqrt{\frac{1}{s}}\delta_d$$

5.3.1 Influence from other Faults

The above discussion is based on the assumption that no faults happen at a same time, so the model (5.16) and (5.17) consider only the influence of local fault and all disturbances on local residual. When there are faults happen at a same time, then the influence from other faults should be considered for node i. For this purpose, set notations

$$\bar{d}_i(k) = \begin{bmatrix} d(k)^T & f_1(k)^T & \cdots & f_{i-1}(k)^T & f_{i+1}(k)^T & \cdots & f_M(k)^T \end{bmatrix}^T,\ \tilde{F}_{di}^* = \begin{bmatrix} \tilde{F}_{di} & 0 \end{bmatrix}.$$

and

$$\tilde{E}_f = \begin{bmatrix} E_{f1} - L_{11}F_{f1} & \cdots & -L_{1(i-1)}F_{f(i-1)} & -L_{1(i+1)}F_{f(i+1)} & \cdots & -L_{1M}F_{fM} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -L_{(i-1)1}F_{f1} & \cdots & E_{f(i-1)} - L_{(i-1)(i-1)}F_{f(i-1)} & -L_{(i-1)(i+1)}F_{f(i+1)} & \cdots & -L_{(i-1)M}F_{fM} \\ -L_{i1}F_{f1} & \cdots & -L_{i(i-1)}F_{f(i-1)} & E_{fi} - L_{i(i+1)}F_{f(i+1)} & \cdots & -L_{iM}F_{fM} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -L_{M1}F_{f1} & \cdots & -L_{M(i-1)}F_{f(i-1)} & -L_{M(i+1)}F_{f(i+1)} & \cdots & E_{fM} - L_{MM}F_{fM} \end{bmatrix}.$$

Substitute d with $\bar{d}_i$, $\tilde{E}_d$ with $\begin{bmatrix} \tilde{E}_d & \tilde{E}_f \end{bmatrix}$, and $\tilde{F}_d$ with $\tilde{F}_d^*$ in (5.16) and (5.17), it holds that

$$\begin{aligned} e(k+1) &= \tilde{A}e(k) + \begin{bmatrix} \tilde{E}_d & \tilde{E}_f \end{bmatrix} \bar{d}_i(k) + \tilde{E}_{fi}f_i(k) \\ r_i(k) &= \tilde{C}_i e(k) + \tilde{F}_d^* \bar{d}_i(k) + F_{fi}f_i(k) \end{aligned}. \tag{5.28}$$

The dynamic relation in (5.28) considers the influence of all disturbance and and fault from other subsystems on the local residual signal in node i. Applying the proposed methods for post-filter based on model (5.28) is a directly extension for designing post-filter to reduce the influence not only from all disturbances but also from faults of other systems on local residual. For the threshold setting, in the fault-free case, we have

$$\|d(k)\|_2 = \|\bar{d}_i(k)\|_2,$$

and the corresponding setting in (5.27) can still be adopted. In this case, although the post-filter is designed to reduce influence of other faults, the happening of other faults may also influence the local residual to cause alarm unless a perfect decoupling is achieved by the designed post-filter.

The algorithms presented in this chapter are summarised in Algorithm 5.1.

Algorithm 5.1. *Distributed FD in interconnected systems based on LMI*

```
Offline Computation
For $i,j = 1, \cdots, M$
Step 1   Compute distributed observer gain $L_{ij}$ based on Theorem 5.2
Step 2   Design post-filter (5.23) based on Theorem 5.3 and 5.4
Step 3   Compute $\gamma_i$ and $J_{th,i} = \gamma_i \sqrt{\frac{1}{s}}\delta_d$ in (5.27)
Online Detection
At time instant $k$, for $i,j = 1, \cdots, M$
Step 1   Build $J_i(k) = \|\epsilon_i(k)\|_{RMS}$ based on (5.2), (5.23), and (5.27)
Step 2   Detect fault using (3.1)
```

5.4 Example

In this section, we provide simulation result to show the effectiveness of the proposed methods. Consider a system model (5.1) consisting of 4 subsystems and the system matrices are

$$A_{11} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}, A_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A_{14} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix},$$

$$A_{21} = \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}, A_{22} = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}, A_{23} = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix},$$

$$A_{32} = \begin{bmatrix} 7 & 0 \\ 0 & 1 \end{bmatrix}, A_{33} = \begin{bmatrix} 3 & 2 \\ 5 & 5 \end{bmatrix}, A_{34} = \begin{bmatrix} 0 & 3 \\ 4 & 5 \end{bmatrix}, C_3 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix},$$

$$A_{41} = \begin{bmatrix} 4 & 7 \\ 0 & 5 \end{bmatrix}, A_{43} = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}, A_{44} = \begin{bmatrix} 1 & 3 \\ 4 & 5 \end{bmatrix}, C_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$E_{d1} = \begin{bmatrix} 0 & -0.1 \\ -0.2 & 0 \end{bmatrix}, F_{d1} = \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.5 \end{bmatrix}, E_{d2} = \begin{bmatrix} 0.3 & 0.2 \\ -0.2 & 0 \end{bmatrix}, F_{d2} = \begin{bmatrix} 0.1 & 0.1 \\ 0.3 & 0 \end{bmatrix},$$

$$E_{d3} = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0 \end{bmatrix}, F_{d3} = \begin{bmatrix} 0.5 & 0 \\ 0.3 & -0.1 \end{bmatrix}, E_{d4} = \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0.5 \end{bmatrix}, F_{d4} = \begin{bmatrix} 0.2 & 0.1 \\ 0.3 & 0 \end{bmatrix}.$$

The topology of the considered system is shown in Figure 5.1. Apply Theorem 5.1 to

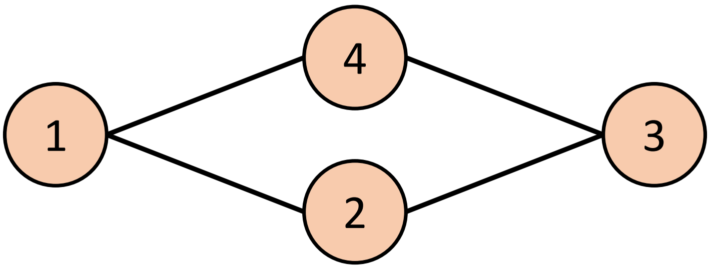


Figure 5.1: System topology

compute the distributed observer gains. The results are

$$L_{11} = \begin{bmatrix} 0.2 & 1 \\ 2.5 & 0.7 \end{bmatrix}, L_{12} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3333 \end{bmatrix}, L_{14} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix},$$

$$L_{21} = \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}, L_{22} = \begin{bmatrix} 1 & 0.3333 \\ 0 & 1 \end{bmatrix}, L_{23} = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix},$$

$$L_{32} = \begin{bmatrix} 3.5 & 0 \\ 0 & 0.3333 \end{bmatrix}, L_{33} = \begin{bmatrix} 3 & 1 \\ 5 & 2.5 \end{bmatrix}, L_{34} = \begin{bmatrix} 0 & 3 \\ 4 & 5 \end{bmatrix},$$

$$L_{41} = \begin{bmatrix} 0.5 & 3.5 \\ -2.5 & 2.5 \end{bmatrix}, L_{43} = \begin{bmatrix} 1 & 1.5 \\ 0 & 1 \end{bmatrix}, L_{44} = \begin{bmatrix} 1 & 3 \\ 4 & 5 \end{bmatrix}.$$

The target for subsystem 1 is to detect local faults in itself with

$$E_{f1} = \begin{bmatrix} 0.1 & 0 \\ 0 & -0.3 \end{bmatrix}, F_{f1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

For the post-filter design, we adopt Theorem 5.3 and 5.4. Set $s = 2$ for model (5.18). We have

$$v_{1,2} = \begin{bmatrix} 0.7982 & -1.1656 & -0.3581 & -1.6689 & 0.4804 & 3.1129 \end{bmatrix}$$

as the result for Theorem 5.3 and

$$G_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, L_1 = -\begin{bmatrix} 0.7982 & -1.1656 \\ -0.3581 & -1.6689 \end{bmatrix}, v_1 = \begin{bmatrix} 0.4804 & 3.1129 \end{bmatrix}, w_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

as the result for Theorem 5.4. Given $\delta_d = 0.45$, further compute $\gamma_{11} = 22.2491$ in (5.26) and set

$$J_1 = \sqrt{\frac{1}{6}\sum_{j=k-5}^{k} \epsilon_1^T(j)\epsilon_1(j)}, \ J_{th,1} = \gamma_1\sqrt{\frac{1}{6}\delta_d} = 4.0874$$

in (5.27). With sampling time $T = 0.01s$, a step fault happens at $30s$, simulation result is shown in Figure 5.2. It can be seen that our proposed method can distributively detect the occurrence of fault.

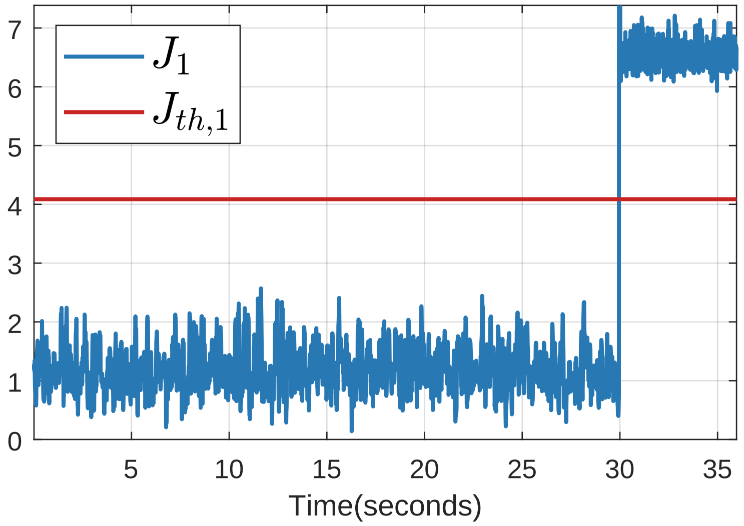


Figure 5.2: Simulation result

5.5 Concluding Remarks

In this chapter, a distributed FD method is designed for large-scale and interconnected systems. Our method is based on a distributed state observer. It uses only its local and neighbours' information to estimate their local states and based on the estimation results to generate the residual signal, which is different between local measurement and its estimation. After this, the residual signal passes through a post-filter to apply for further local FD. The design of the post-filter considers the trade-off between robustness to disturbance and faults form other subsystems, and the sensitivity to local fault. For the design of the corresponding parameters, LMI techniques are implemented for the design of distributed observer gains to guarantee the convergence of residual signals and also the design of post-filter. A combination of PSA and DO is also proposed as an analytical solution for post-filter design. Finally, how to reduce the influence of other fault on local detection performance is also discussed in Section 5.3.1. Notice that although the LMI technique offers a convenient approach to address complex design problems, it provides only sufficient conditions for optimization. And although it is considered to reduce the influence of other faults, the local residual may also be influenced by faults from other systems unless the post-filter achieves a perfect decoupling.

6 Distributed Fault Detection in Interconnected Systems via Optimal Estimation

The methods outlined in Chapters 4 and 5 apply distributed average consensus or distributed state observer techniques to achieve FD in large-scale and interconnected systems through the use of sensor networks. To compute the residual signal at each time instant, the former method utilizes, in fact, information from all nodes in the network, while the latter method only uses local and neighbours' information. This observation inspires us to propose a novel approach for selecting pertinent information that is effective and efficient in achieving FD in a given system. This thinking and its comparison with former methods are shown in Figure 6.1.

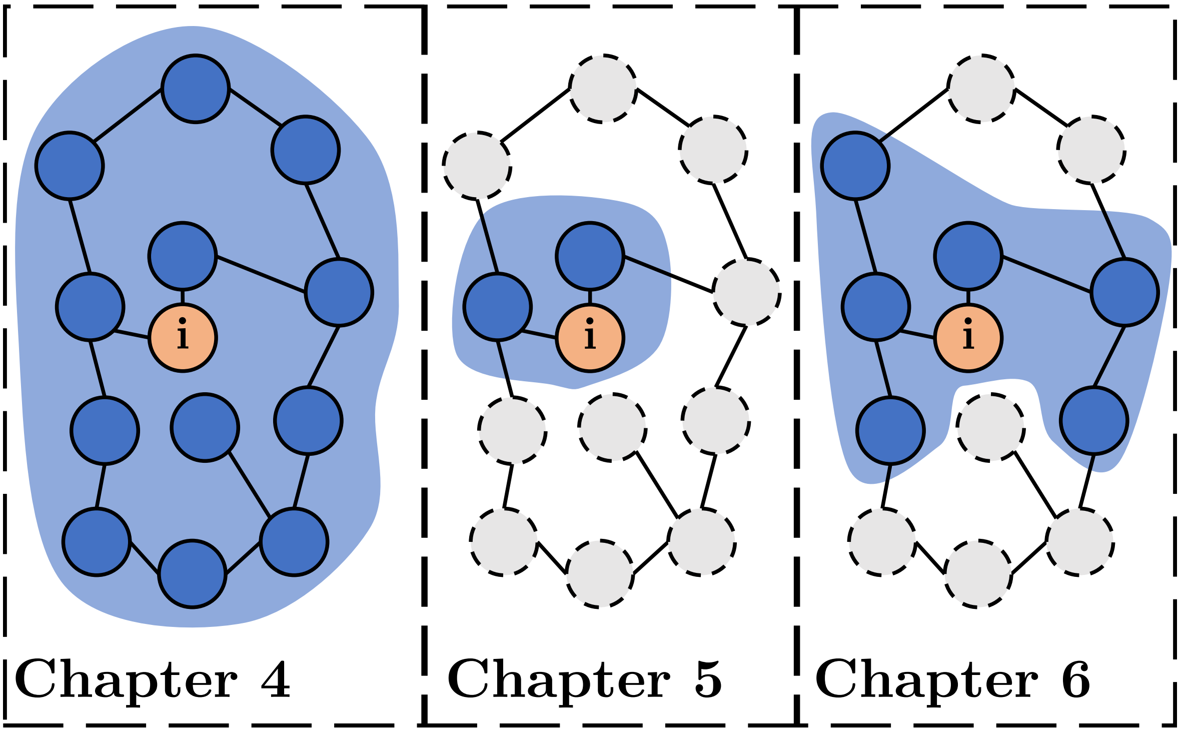


Figure 6.1: Information involved in different methods

Based on the aforementioned discussion, the primary aim of this chapter is to establish a distributed approach for FD in large-scale and interconnected systems through the utilization of a sensor network. Initially, a criterion will be established to select which information is important to FD. Subsequently, the distributed implementation of an online FD strategy shall be introduced based on the chosen information. We apply the method for both static and dynamic systems. Moreover, the proposed method also considers the transmission time of information among sensor network and is applied to both static and dynamic cases.

Notations: Besides the notations used in Chapter 2 for graph theory, denote $d_i = \max_{j\in\mathcal{N}} d(i,j)$ as the maximal distance of node i among the graph and $\mathcal{N}_{i,\rho} = \{j|\ d(i,j) = \rho\}$ as the $\rho-$layer neighbours of node i. Especially, we have $\mathcal{N}_{i,0} = \{i\}$ and $\mathcal{N}_{i,1} = \mathcal{N}_i$. For our purpose, each set is arranged in ascending order of the values with $\mathcal{N}_{i,j} = \{n_{i,j}^1, n_{i,j}^2, \cdots, n_{i,j}^{c_{i,j}}\}$, where $c_{i,j} = \text{card}(\mathcal{N}_{i,j})$. When a signal y_i is related to node i, define $y_{\mathcal{N}_{i,j}}$ as the collection of signal from the $j-$layer neighbours, that is,

$$y_{\mathcal{N}_{i,j}} = \begin{bmatrix} y_{n_{i,j}^1}^T & y_{n_{i,j}^2}^T & \cdots & y_{n_{i,j}^{c_{i,j}}}^T \end{bmatrix}^T. \tag{6.1}$$

For example of the above notations in Figure 6.2, we have for node 1,

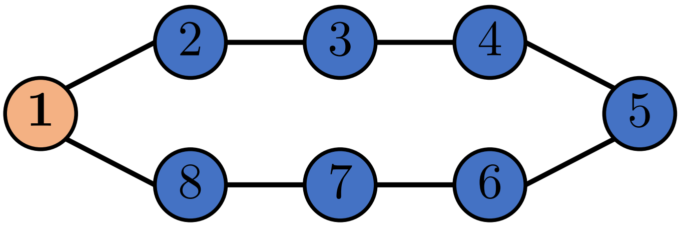


Figure 6.2: An example for new notations

$$d_1 = 4, \mathcal{N}_{1,0} = \{1\}, \mathcal{N}_{1,1} = \{2,8\}, \mathcal{N}_{1,2} = \{3,7\}, \mathcal{N}_{1,3} = \{4,6\}, \mathcal{N}_{1,4} = \{5\},$$

$$y_{\mathcal{N}_{1,0}} = \begin{bmatrix} y_1 \end{bmatrix}, y_{\mathcal{N}_{1,1}} = \begin{bmatrix} y_2 \\ y_8 \end{bmatrix}, y_{\mathcal{N}_{1,2}} = \begin{bmatrix} y_3 \\ y_7 \end{bmatrix}, y_{\mathcal{N}_{1,3}} = \begin{bmatrix} y_4 \\ y_6 \end{bmatrix}, y_{\mathcal{N}_{1,4}} = \begin{bmatrix} y_5 \end{bmatrix}.$$

For matrix representation, when matrix A is partitioned into $m \times m$ blocks, denote A_{ij} as the $(i,j)-$th block of matrix A and $\text{e}[A]_{k,q}$ as the submatrix of A by extracting the first $k \times q$ blocks of A. Denote $\text{c}(B)_{i,k} = \begin{bmatrix} 0 & \cdots & B^T & \cdots & 0 \end{bmatrix}^T$ as k-block-column matrix with B located in the i-th row block and other elements 0. For example,

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \ \text{e}[A]_{2,1} = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}, \text{c}(B)_{1,2} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \ \text{c}(B)_{2,3} = \begin{bmatrix} 0 \\ B \\ 0 \end{bmatrix}.$$

6.1 Preliminaries and Problem Formulation for Static Systems

First, we focus on the static case. This section provides an introduction to the model of a sensor network, optimal FD algorithm, and problem formulation for static scenarios.

6.1.1 Model Description

For the purpose of monitoring large-scale industrial systems, the process is equipped with a sensor network constituted of N nodes. Each node stands for a local sensor, has sampling time T, and is modelled by

$$y_i(k) = A_i x + B_i f_i(k) + \varepsilon_i(k), \tag{6.2}$$

where node index $i \in \{1, \cdots, N\}$, $y_i \in \mathcal{R}^{m_{y_i}}$ is the measurement of node i, A_i is the measurement matrix, $x \in \mathcal{R}^n$ is the state vector of the large-scale system, B_i is the known distribution matrix of f_i with full column rank, $f_i \in \mathcal{R}^{m_{f_i}}$ is the sensor fault at node i which equals zero when in fault-free case otherwise doesn't equal zero, and ε_i stands for the local measurement noise with

$$\varepsilon \sim \mathcal{N}(0, \Sigma),\ \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_N \end{bmatrix},\ \Sigma = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1N} \\ \vdots & \ddots & \vdots \\ \Sigma_{N1} & \cdots & \Sigma_{NN} \end{bmatrix}.$$

It is assumed that node can only communicate with their neighbours ($1-$layer neighbours) with a (unit) communication time T. From this assumption, the transmission of information between nodes $j \in \mathcal{N}_{i,k}$ and node i is indirect and costs kT.

6.1.2 An Optimal Fault Detection Scheme

For model (6.2), define $r_i(k) = y_i(k) - A_i x$, it holds that

$$r_i(k) = B_i f_i(k) + \varepsilon_i(k). \tag{6.3}$$

Assume that multiple faults do not occur simultaneously. Following the study in Section 3.2.1 and Theorem 3.1, the detection problem of f_i on the assumption of the model (6.3) can be brought into the following global form

$$r(k) = \bar{B}_i f_i(k) + \varepsilon(k), \tag{6.4}$$

where

$$r(k) = \begin{bmatrix} r_1^T(k) & r_2^T(k) & \cdots & r_N^T(k) \end{bmatrix},\ \bar{B}_i = \mathrm{c}(B_i)_{i,\mathcal{N}}. \tag{6.5}$$

Since B_i is full column rank matrix, so is $\bar{B}_i$. And the solution is given by

$$J_i(k) = r^T(k)\Sigma^{-1}\bar{B}_i(\bar{B}_i^T\Sigma^{-1}\bar{B}_i)^{-1}\bar{B}_i^T\Sigma^{-1}r(k), \ J_{i,th} = \chi^2_\alpha(m_{f_i}) \tag{6.6}$$

6.1.3 Problem Formulation

Since the transmission of measurements from node $j \in \mathcal{N}_{i,\rho}$ to node i costs the same time, we rearrange the nodes according to the $d(i,j)$ and separate it into d_i parts as

$$\bar{r}_i = \begin{bmatrix} r^T_{\mathcal{N}_{i,0}} & r^T_{\mathcal{N}_{i,1}} & \cdots & r^T_{\mathcal{N}_{i,d_i}} \end{bmatrix}^T, \tag{6.7}$$

Notice that (6.7) rearranges the nodes by permuting the row partitions of r, and R_i is a permutation matrix of node i and thus invertible. Due to the assumption of transmission time, at time index k, $r_{\mathcal{N}_{i,0}}(k)$, $r_{\mathcal{N}_{i,1}}(k-1)$, $\cdots$, $r_{\mathcal{N}_{i,d_i}}(k-d_i)$ are obtained by node i. Also, the computation of (6.6) can only be achieved at $(k+d_i)$ and thus has MTTD d_i. To reduce the MTTD of f_i in node i and meanwhile maintain the performance of FD to a certain level, we investigate a distributed FD scheme for static system (6.2) in the next section.

6.2 A Distributed Fault Detection Scheme for Static Systems

In this section, we propose a distributed FD scheme to realize the optimal solution (6.6) in a distributed manner and meanwhile to make a trade-off between the FD performance and its MTTD.

6.2.1 Node Selection

First, consider what node i can benefit from more information. Intuitively, more information can improve the performance of FD by reducing uncertainty. In this section, the improvement from more information is quantified. Rearrange and separate r in model (6.4) into two parts r_i and the rest residuals as

$$\bar{r} = \begin{bmatrix} r_i^T & r_R^T \end{bmatrix}^T, \ r_R = \begin{bmatrix} r_1^T & \cdots & r_{i-1}^T & r_{i+1}^T & \cdots & r_N^T \end{bmatrix}^T,$$

matrix $\bar{B}_i$ is correspondingly changed to $\begin{bmatrix} B_i^T & 0 \end{bmatrix}^T$ to build (6.6) and

$$\bar{r} \sim \mathcal{N}\left(0, \bar{\Sigma}\right), \ \bar{\Sigma} = \begin{bmatrix} \Sigma_{ii} & \Sigma_{iR} \\ \Sigma_{Ri} & \Sigma_{RR} \end{bmatrix}.$$

Set parameters

$$\bar{r}_i = r_i - \Sigma_{iR}\Sigma_{RR}^{-1}r_R,\ \bar{\Sigma}_{ii} = \Sigma_{ii} - \Sigma_{iR}\Sigma_{RR}^{-1}\Sigma_{Ri}. \tag{6.8}$$

Based on (6.8) and the following identity

$$\begin{bmatrix} I & -\Sigma_{iR}\Sigma_{RR}^{-1} \\ 0 & I \end{bmatrix} \bar{\Sigma} \begin{bmatrix} I & 0 \\ -\Sigma_{RR}^{-1}\Sigma_{Ri} & I \end{bmatrix} = \begin{bmatrix} \bar{\Sigma}_{ii} & 0 \\ 0 & \Sigma_{RR} \end{bmatrix},$$

evaluation function in (6.6) is reformulated as

$$\begin{aligned} J_i(k) =& \bar{r}^T(k)\bar{\Sigma}^{-1}\begin{bmatrix} B_i \\ 0 \end{bmatrix}(\begin{bmatrix} B_i \\ 0 \end{bmatrix}^T\bar{\Sigma}^{-1}\begin{bmatrix} B_i \\ 0 \end{bmatrix})^{-1}\begin{bmatrix} B_i \\ 0 \end{bmatrix}^T\bar{\Sigma}^{-1}\bar{r} \\ =& \bar{r}_i^T(k)\bar{\Sigma}_{ii}^{-1}B_i(B_i^T\bar{\Sigma}_{ii}^{-1}B_i)^{-1}B_i^T\bar{\Sigma}_{ii}^{-1}\bar{r}_i \end{aligned}. \tag{6.9}$$

If only measurement from node i is used for detection, the optimal evaluation function is

$$\bar{J}_i(k) = r_i^T(k)\Sigma_{ii}^{-1}B_i(B_i^T\Sigma_{ii}^{-1}B_i)^{-1}B_i^T\Sigma_{ii}^{-1}r_i. \tag{6.10}$$

According to Lemma 2.2, $\bar{r}_i = r_i - \Sigma_{iR}\Sigma_{RR}^{-1}r_R$ stands for the difference between r_i and its orthogonal projection-based estimation $\Sigma_{iR}\Sigma_{RR}^{-1}r_R$ and $\bar{\Sigma}_{ii}$ delivers the minimal variance matrix of the estimation error $\bar{r}_i$. Compare (6.9) and (6.10), signal r_i is transferred into $\bar{r}_i$ with extra measurement and the uncertainty, which is represented by variance matrix, is minimized from Σ_{ii} to $\bar{\Sigma}_{ii}$. For further analysis, we first introduce the following lemma.

Lemma 6.1. *Given*

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} & \Sigma_{xz} \\ \Sigma_{yx} & \Sigma_y & \Sigma_{yz} \\ \Sigma_{zx} & \Sigma_{zy} & \Sigma_z \end{bmatrix}\right),$$

From Lemma 2.2, the optimal estimation of x given y and (y, z) are

$$\hat{x}_y = \Sigma_{xy}\Sigma_{yy}^{-1}y,\ \hat{x}_{y,z} = \begin{bmatrix} \Sigma_{xy} & \Sigma_{xz} \end{bmatrix}\begin{bmatrix} \Sigma_y & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_z \end{bmatrix}^{-1}\begin{bmatrix} y \\ z \end{bmatrix},$$

respectively. It holds that

$$\mathrm{var}(x - \hat{x}_y) \geqslant \mathrm{var}(x - \hat{x}_{y,z}).$$

Proof. It is straightforward that

$$\begin{aligned} \mathrm{var}(x - \hat{x}_y) &= \Sigma_x - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}, \\ \mathrm{var}(x - \hat{x}_{y,z}) &= \Sigma_x - \begin{bmatrix} \Sigma_{xy} & \Sigma_{xz} \end{bmatrix}\begin{bmatrix} \Sigma_y & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_z \end{bmatrix}^{-1}\begin{bmatrix} \Sigma_{yx} \\ \Sigma_{zx} \end{bmatrix}. \end{aligned}$$

Define notations

$$\bar{\Sigma}_{yz} = -\Sigma_y^{-1}\Sigma_{yz},\ \bar{\Sigma}_z = \Sigma_z - \Sigma_{zy}\Sigma_y^{-1}\Sigma_{yz},\ \bar{\Sigma}_{xz} = \Sigma_{xz} + \Sigma_{xy}\bar{\Sigma}_{yz},$$

and apply

$$\begin{bmatrix} \Sigma_y & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_z \end{bmatrix}^{-1} = \begin{bmatrix} I & \bar{\Sigma}_{yz} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_y & 0 \\ 0 & \bar{\Sigma}_z \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ \bar{\Sigma}_{yz}^T & I \end{bmatrix}$$

to modify $\mathrm{var}(x - \hat{x}_{y,z})$ into

$$\mathrm{var}(x - \hat{x}_{y,z}) = \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx} - \bar{\Sigma}_{xz}\bar{\Sigma}_z^{-1}\bar{\Sigma}_{xz}^T.$$

Since $\bar{\Sigma}_{xz}\bar{\Sigma}_z^{-1}\bar{\Sigma}_{xz}^T \geqslant 0$, Lemma 6.1 is proved. □

From Lemma 6.1, if more additional measurement is used for detection, the variance matrix of the estimation error becomes even smaller, that is, the more measurement, the stronger influence of uncertainty can be reduced. Since the transmission of information from node $j \in \mathcal{N}_{i,\rho}$ to node i costs same time, we select nodes layer by layer, that is, given the accepted level of uncertainty, if the information inside k-th layer is enough, then the information from $(k+1)$-th layer to d_i-th layer are not further considered for FD. For this purpose, define variance matrix of $\bar{r}_i$ in (6.7) as

$$\Sigma_i = \mathrm{var}(\bar{r}_i) = \begin{bmatrix} \Sigma_{0,0} & \cdots & \Sigma_{0,d_i} \\ \vdots & \ddots & \vdots \\ \Sigma_{d_i,0} & \cdots & \Sigma_{d_i,d_i} \end{bmatrix}, \tag{6.11}$$

where $\Sigma_{m,n} = \mathrm{cov}(r_{\mathcal{N}_{i,m}}, r_{\mathcal{N}_{i,n}})$. Define $\widetilde{\Sigma}_{i,\rho}$, $\rho = 0, \cdots, d_i$, as the variance matrix of the residual signals between r_i and its optimal estimation by projecting itself into the subspace spanned by $\{r_{\mathcal{N}_{i,1}}, \cdots, r_{\mathcal{N}_{i,\rho}}\}$. It holds that

$$\widetilde{\Sigma}_{i,\rho} = \Sigma_{0,0} - \begin{bmatrix} \Sigma_{0,1} & \cdots & \Sigma_{0,\rho} \end{bmatrix} \begin{bmatrix} \Sigma_{1,1} & \cdots & \Sigma_{1,\rho} \\ \vdots & \ddots & \vdots \\ \Sigma_{\rho,1} & \cdots & \Sigma_{\rho,\rho} \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_{1,0} \\ \vdots \\ \Sigma_{\rho,0} \end{bmatrix}.$$

And the problem of node selection for node i is formulated as

$$\min_{0 \leqslant \rho \leqslant d_i} \rho,\ \text{s.t. } \mathrm{tr}(\widetilde{\Sigma}_{i,\rho}) \leqslant \delta, \tag{6.12}$$

where the matrix trace is used to measure the uncertainty with $\delta \geqslant 0$ as its given bound.

6.2.2 A Distributed Fault Detection Scheme

After selecting the ρ_i as the solution of (6.12) for node i, the centralized FD of $f_i(k)$ is achieved at $k + \rho_i$. However, at time index k, the information about $f_i(k)$ is already obtained, so the FD of $f_i(k)$ can be applied from k on. Motivated by this, the iterative distributed realization of optimal detection is introduced in this section. First, the following lemma about the diagonalization of the variance matrix is introduced.

Lemma 6.2. *Given*

$$x \sim \mathcal{N}(0, \Sigma),\ x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix},\ \Sigma = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1N} \\ \vdots & \ddots & \vdots \\ \Sigma_{N1} & \cdots & \Sigma_{NN} \end{bmatrix}.$$

For $k = 1, \cdots, N-1$, define

$$\Sigma(1) = \Sigma,\ \Sigma(k+1) = P(k)\Sigma(k)P^{\top}(k), \tag{6.13}$$

$P(k)$ with its (i,j)-th block as

$$P_{ij}(k) = \begin{cases} I & , i = j \\ -\Sigma_{ij}(k)\Sigma_{jj}^{-1}(k) & , i = k+1, j < k+1 \\ 0 & , \text{others} \end{cases} \tag{6.14}$$

and $P^{[k]} = \prod_{i=k}^{1} P(i)$. Further, for $1 \leqslant s \leqslant N$,

$$x_s = \begin{bmatrix} x_1 \\ \vdots \\ x_s \end{bmatrix} \sim \mathcal{N}(0, \Sigma_s),\ \Sigma_s = \mathrm{e}[\Sigma]_{s,s} = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1s} \\ \vdots & \ddots & \vdots \\ \Sigma_{s1} & \cdots & \Sigma_{ss} \end{bmatrix}.$$

For $q = 1, \cdots, s-1$, define

$$\Sigma_s(1) = \Sigma_s,\ \Sigma_s(q+1) = P_s(q)\Sigma_s(q)P_s^T(q),$$

$P_s(q)$ with its (i,j)-th block

$$P_{s,ij}(q) = \begin{cases} I & , i = j \\ -\Sigma_{s,ij}(q)\Sigma_{s,jj}^{-1}(q) & , i = q+1, j < q+1 \\ 0 & , \text{others} \end{cases}$$

and $P_s^{[q]} = \prod_{i=q}^{1} P_s(i)$. *It holds that*

$$P^{[k]}x \sim \mathcal{N}(0, \Sigma(k+1)),\ \ P_s^{[q]}x_s \sim \mathcal{N}(0, \Sigma_s(q+1)) \tag{6.15}$$

$$\Sigma(k) = \begin{bmatrix} \xi(k) & \star \\ \star & \star \end{bmatrix},\ \xi(k) = \begin{bmatrix} \bar{\Sigma}_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \bar{\Sigma}_{kk} \end{bmatrix} \tag{6.16}$$

$$\mathrm{e}[P^{[k]}]_{k+1,k+1} = \mathrm{e}[P^{[N-1]}]_{k+1,k+1} \tag{6.17}$$

$$P_s^{[q]} = \mathrm{e}[P^{[q]}]_{s,s} \tag{6.18}$$

$$P_s(q) = \mathrm{e}[P(q)]_{s,s} \tag{6.19}$$

$$\Sigma_s(q+1) = \mathrm{e}[\Sigma(q+1)]_{s,s} \tag{6.20}$$

Proof. The statement (6.15) is obvious. We apply mathematical induction to prove statement (6.16). For $2 \leqslant k \leqslant N-1$, set

$$\begin{gathered} x(1) = x, x(k) = P(k-1)x(k-1), \bar{\Sigma}_{11} = \Sigma_{11}(1), \\ D(k) = \begin{bmatrix} \Sigma_{(k+1)k}(k) & \cdots & \Sigma_{(k+1)k}(k) \end{bmatrix}, \xi(k) = \begin{bmatrix} \bar{\Sigma}_{11} & & 0 \\ & \ddots & \\ 0 & & \bar{\Sigma}_{kk} \end{bmatrix}, \bar{\Sigma}_{kk} = \Sigma_{kk}(k), \end{gathered} \tag{6.21}$$

where $\Sigma(k)$ is as defined in (6.13). It holds that

$$\mathrm{var}(x(k)) = \Sigma(k),\ \mathrm{var}(x(1)) = \Sigma(1) = \Sigma.$$

For $k = 2$ and according to (6.14), construct

$$P(1) = \begin{bmatrix} p(1) & 0 \\ 0 & I \end{bmatrix},\ p(1) = \begin{bmatrix} I & 0 \\ -\Sigma_{21}(1)\Sigma_{11}^{-1}(1) & I \end{bmatrix}.$$

It is straightforward that

$$\Sigma(2) = P(1)\Sigma(1)P(1)^T = \begin{bmatrix} \xi(2) & \star \\ \star & \star \end{bmatrix}, \xi(2) = \begin{bmatrix} \bar{\Sigma}_{11} & 0 \\ 0 & \bar{\Sigma}_{22} \end{bmatrix},$$

where $\bar{\Sigma}_{11}$ is as shown in (6.21) and

$$\bar{\Sigma}_{22} = \Sigma_{22}(1) - \Sigma_{21}(1)\bar{\Sigma}_{11}^{-1}\Sigma_{12}(1).$$

Assume at $k = l$, it holds that

$$\Sigma(l) = \begin{bmatrix} \xi(l) & D^T(l) & \star \\ D(l) & \Sigma_{(l+1)(l+1)}(l) & \star \\ \star & \star & \star \end{bmatrix}, \xi(l) = \begin{bmatrix} \bar{\Sigma}_{11} & & 0 \\ & \ddots & \\ 0 & & \bar{\Sigma}_{ll} \end{bmatrix}$$

Define $C(l) = -D(l)\xi(l)^{-1}$. Compute $P(l)$ using (6.14) and

$$\Sigma(l+1) = \begin{bmatrix} I & 0 & 0 \\ C(l) & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \xi(l) & D^T(l) & \star \\ D(l) & \Sigma_{(l+1)(l+1)}(l) & \star \\ \star & \star & \star \end{bmatrix} \begin{bmatrix} I & C^T(l) & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} = \begin{bmatrix} \xi(l+1) & \star \\ \star & \star \end{bmatrix},$$

where

$$\xi(l+1) = \begin{bmatrix} \xi(l) & 0 \\ 0 & \bar{\Sigma}_{(l+1)(l+1)} \end{bmatrix}, \bar{\Sigma}_{(l+1)(l+1)} = \Sigma_{(l+1)(l+1)}(l) - D(l)\xi(l)^{-1}D^T(l).$$

Thus statement (6.16) is proved. For $k+1 \leqslant q \leqslant N-1$,

$$\mathrm{e}[P(q)]_{k+1,N} = \begin{bmatrix} I & 0 \end{bmatrix}. \tag{6.22}$$

Since (6.22) and $P = P(N-1)\cdots P(k+1)P^{[k]}$, the first $(k+1)\times(k+1)$ block submatrices of $P^{[k]}$ and P are the same, thus the statement (6.17) is proved. Define $\bar{x}_1 = x_1$ and

$$\bar{x}_k = x_k - \sum_{j=1}^{k-1} \Sigma_{kj}(k-1)\bar{\Sigma}_{jj}^{-1}\bar{x}_j. \tag{6.23}$$

Combining (6.15), (6.16), (6.21), and (6.23), it holds that

$$x(k) = \begin{bmatrix} \bar{x}_1^T & \cdots & \bar{x}_k^T & x_{k+1}^T & \cdots & x_N^T \end{bmatrix}^T, \tag{6.24}$$

where $x(k)$ is defined in (6.21), and

$$\mathrm{var}\left(\begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_k \end{bmatrix}\right) = \xi(k), \ \mathrm{cov}\left(x_{k+1}, \begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_k \end{bmatrix}\right) = D(k).$$

Based on above properties, reformulate (6.23) into

$$\bar{x}_k = x_k - \hat{x}_k, \ \hat{x}_k = D(k-1)\xi(k-1)^{-1} \begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_{k-1} \end{bmatrix} \tag{6.25}$$

From (6.21) and (6.24), matrix $P(k-1)$ transfer x_k into $\bar{x}_k$. This transformation, as shown in (6.25) and according to Lemma 2.2, is achieved by building residual between x_k and its optimal estimation $\hat{x}_k$, which is obtained by projecting x_k into the subspace spanned by $\{\bar{x}_1, \cdots, \bar{x}_{k-1}\}$. Moreover, since P is invertible, the subspace spanned by $\{\bar{x}_1, \cdots, \bar{x}_{k-1}\}$ is the same one spanned by $\{x_1, \cdots, x_{k-1}\}$. Because the generation of $\bar{x}_k$ is based on $x_1, \cdots, x_{k-1}$ and the statistic properties among them, the variables $x_{k+1}, \cdots, x_N$ have no influence on this diagonalization. Based on this, statement (6.18), (6.19), and (6.20) are proved. □

Lemma 6.2 shows a way to diagonalize variance matrix Σ with $P = P^{[N-1]}$, a procedure to diagonalize the submatrix variance matrix Σ_s with $P_s = P_s^{[s-1]}$ and the relation between them $P_s = \mathrm{e}[P]_{s,s}$. A similar procedure is introduced in [64] and is known as canonical covariance factorization.

Theorem 6.1. *Given model (6.2), for node i, rearrange the sequence of nodes according to (6.7) to obtain $\bar{r}_i$ and corresponding variance matrix Σ_i as defined in (6.11). Compute lower triangular block matrices*

$$P_i(l) = \begin{bmatrix} I & 0 & \cdots & 0 \\ P_{i,1,0}(l) & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ P_{i,d_i,0}(l) & \cdots & P_{i,d_i,d_i-1}(l) & I \end{bmatrix}, P_i = \prod_{l=d_i}^{1} P_i(l) = \begin{bmatrix} I & 0 & \cdots & 0 \\ P_{i,1,0} & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ P_{i,d_i,0} & \cdots & P_{i,d_i,d_i-1} & I \end{bmatrix} \tag{6.26}$$

according to Lemma 6.2 to diagonalize Σ_i with

$$P_i \Sigma_i P_i^T = \bar{\Sigma}_i = \begin{bmatrix} \bar{\Sigma}_{i,0} & & & 0 \\ & \bar{\Sigma}_{i,1} & & \\ 0 & & \ddots & \\ & & & \bar{\Sigma}_{i,d_i} \end{bmatrix} \tag{6.27}$$

Solve (6.12) to obtain ρ_i for node selection. For $s = 1, \cdots, \rho_i$, run

$$Q_{i,0} = B_i^T \bar{\Sigma}_0^{-1} B_i \tag{6.28}$$

$$Q_{i,s} = Q_{i,s-1} + B_i^T P_{i,s,0}^T \bar{\Sigma}_{i,s}^{-1} P_{i,s,0} B_i \tag{6.29}$$

$$\bar{r}_{i,0}(k) = r_{\mathcal{N}_{i,0}}(k) \tag{6.30}$$

$$\bar{r}_{i,s}(k) = r_{\mathcal{N}_{i,s}}(k) + \sum_{m=1}^{s-1} P_{i,s,m}(s) \bar{r}_{i,m}(k) \tag{6.31}$$

$$\widetilde{r}_{i,0}(k) = B_i^T \bar{\Sigma}_{i,0}^{-1} \bar{r}_{i,0}(k) \tag{6.32}$$

$$\widetilde{r}_{i,s}(k) = \widetilde{r}_{i,s-1}(k) + B_i^T P_{i,s,0}^T \bar{\Sigma}_{i,s}^{-1} \bar{r}_{i,s}(k) \tag{6.33}$$

And the evaluation functions

$$J_{i,k}(k) = \widetilde{r}_{i,0}^T(k) Q_{i,0}^{-1} \widetilde{r}_{i,0}(k) \tag{6.34}$$

$$J_{i,k+s}(k) = \widetilde{r}_{i,s}^T(k) Q_{i,s}^{-1} \widetilde{r}_{i,s}(k) \tag{6.35}$$

$$J_{i,th} = \chi_\alpha^2(m_{f_i}) \tag{6.36}$$

deliver the distributed solution to (6.6).

Proof. For $0 \leqslant s \leqslant \rho_i$, define $B_{i,s} = \mathrm{c}(B_i)_{1,s+1}$,

$$q_{i,s} = \begin{bmatrix} r_{\mathcal{N}_{i,0}} \\ \vdots \\ r_{\mathcal{N}_{i,s}} \end{bmatrix}, \ \Sigma_{i,s} = \begin{bmatrix} \Sigma_{0,0} & \cdots & \Sigma_{0,s} \\ \vdots & \ddots & \vdots \\ \Sigma_{s,0} & \cdots & \Sigma_{s,s} \end{bmatrix}.$$

From (6.6), it holds that

$$J_{i,k+s}(k) = q_{i,s}^T(k)\Sigma_{i,s}^{-1}B_{i,s}(B_{i,s}^T\Sigma_{i,s}^{-1}B_{i,s})^{-1}B_{i,s}^T\Sigma_{i,s}^{-1}q_{i,s}(k), \ J_{i,th} = \chi_\alpha^2(m_{f_i}) \tag{6.37}$$

deliver the optimal FD for $f_i(k)$ at time $k+s$. When $s=0$, it is straightforward that (6.28), (6.30), (6.32) and (6.34) hold. According to Lemma 6.2, we have

$$\mathrm{e}[\bar{\Sigma}_i]_{s+1,s+1} = \mathrm{e}[P_i]_{s+1,s+1}(\Sigma_{i,s})\mathrm{e}[P_i]_{s+1,s+1}^T. \tag{6.38}$$

From (6.38),it is straightforward that

$$Q_{i,s} = B_{i,s}^T\Sigma_{i,s}^{-1}B_{i,s} = B_{i,s}^T\mathrm{e}[P_i]_{s+1,s+1}^T(\mathrm{e}[\bar{\Sigma}_i]_{s+1,s+1})^{-1}\mathrm{e}[P_i]_{s+1,s+1}B_{i,s}$$
$$= Q_{i,0} + \sum_{n=1}^{s} B_i^T P_{i,n,0}^T\bar{\Sigma}_{i,n}^{-1}P_{i,n,0}B_i = Q_{i,s-1} + B_i^T P_{i,s,0}^T\bar{\Sigma}_{i,s}^{-1}P_{i,s,0}B_i,$$

thus (6.29) holds. Further, with identity

$$\begin{bmatrix} \bar{r}_{i,0}(k)^T & \cdots & \bar{r}_{i,s}(k)^T \end{bmatrix}^T = \mathrm{e}[P_i]_{s+1,s+1}q_{i,s}(k) = \prod_{i=s}^{1}\mathrm{e}[P_i(i)]_{s+1,s+1}q_{i,s}(k),$$

equation (6.31) holds and based on this, we have

$$\widetilde{r}_{i,s}(k) = B_{i,s}^T\Sigma_{i,s}^{-1}q_{i,s}(k) = B_{i,s}^T\mathrm{e}[P_i]_{s+1,s+1}^T(\mathrm{e}[\bar{\Sigma}_i]_{s+1,s+1})^{-1}\mathrm{e}[P_i]_{s+1,s+1}q_{i,s}(k)$$
$$= \widetilde{r}_{i,0}(k) + \sum_{n=1}^{s} B_i^T P_{i,n,0}^T\bar{\Sigma}_{i,n}^{-1}\bar{r}_{i,n}(k) = \widetilde{r}_{i,s-1}(k) + B_i^T P_{i,s,0}^T\bar{\Sigma}_{i,s}^{-1}\bar{r}_{i,s}(k),$$

equation (6.33) is proved. From (6.37), it is obvious that (6.35) hold. And the threshold setting (6.36) is directly from Lemma 3.1. □

Through the method proposed in Theorem 6.1, at $k+s$, $\bar{r}_{i,s}(k)$ in (6.31) is the residual signal between $r_{\mathcal{N}_{i,s}}(k)$ and its optimal estimation by projecting itself into the subspace spanned by $\{\bar{r}_{i,0}(k), \cdots, \bar{r}_{i,s-1}(k)\}$ and this residual $\bar{r}_{i,s}(k)$ is applied for improving the accuracy of FD at $k+s$. In such a way, from k to $k+\rho_i$, the optimal detection of $f(k)$ is achieved in a distributed and recursive form. The overall algorithm is summarized in Algorithm 6.1.

Algorithm 6.1. *Distributed FD for static systems*

```
Offline Computation
For $i = 1, \cdots, N$
Step 1  At node $i$, obtain fault-free offline data to train mean of $A_i x$ and variance $\Sigma$
Step 2  Solve (6.12) to obtain $\rho_i$
Step 3  Calculate $P_i(m)$, $P_i$ and $\bar{\Sigma}_i$ in (6.26) and (6.27)
Step 4  Calculate $Q_{i,0}, Q_{i,1}, \cdots, Q_{i,\rho_i}$ in (6.28) and (6.29)
Step 5  Choose $\alpha$ as FAR and calculate $J_{i,th}$ in (6.36)
Online Detection
For $i = 1, \cdots, N$ and $s = 1, \cdots, \rho_i$
Step 1  At $k$, obtain $y_i(k)$, build $r_i(k) = y_i(k) - A_i x$, calculate $\bar{r}_{i,0}(k)$ and $\widetilde{r}_{i,0}(k)$ in
        (6.30) and (6.32), build $J_{i,k}(k)$ in (6.34) and apply (3.1) to make decision
Step 2  At $(k+s)$, obtain $r_{\mathcal{N}_{i,s}}(k)$, calculate $\bar{r}_{i,s}(k)$ and $\widetilde{r}_{i,s}(k)$ using (6.31) and (6.34),
        build $J_{i,k+s}(k)$ in (6.35) and apply (3.1) to make decision
```

6.3 Preliminaries and Problem Formulation for Dynamic Sytems

Now, we shift our focus to the dynamic case. This section aims to introduce the models of the dynamic system, centralized FD algorithm, and problem formulation.

6.3.1 Model Description

The large-scale interconnected system under consideration consists of N subsystems with the same sampling time T. Each subsystem G_i, $i = 1, \cdots, N$, is modelled by

$$x_i(k+1) = A_{ii}x_i(k) + \sum_{j \in \mathcal{N}_i} A_{ij}x_j(k) + w_i(k), \;\; y_i(k) = C_i x_i(k) + v_i(k), \tag{6.39}$$

where $x_i(k) \in \mathcal{R}^{n_i}$ is the local state, $y_i(k) \in \mathcal{R}^{p_i}$ is the local measurement, $w_i(k) \in \mathcal{R}^{n_i}$ and $v_i(k) \in \mathcal{R}^{p_i}$ denote the process and measurement noise, respectively. We assume that, for $i, j = 1, \cdots, N$ and $k, h \geqslant 0$, $w_i(k)$ and $v_i(k)$ are Gaussian white noises, the initial state $x_i(0)$ is random variables following a Gaussian distribution and is uncorrelated with process and measurement noises.

$$\begin{gathered} x_i(0) \sim \mathcal{N}(\bar{x}_{i,0}, \Sigma_i), \;\; \operatorname{cov}\left(x_i(0), x_j^T(0)\right) = \Sigma_i \delta_{ij} \\ \operatorname{cov}\left(\begin{bmatrix} w_i(k) \\ v_i(k) \end{bmatrix}, \begin{bmatrix} w_j^T(h) & v_j^T(h) \end{bmatrix} \right) = \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix} \delta_{kh}\delta_{ij} \end{gathered} \tag{6.40}$$

Matrix $A_{ij} \neq 0$, when $j \in \mathcal{N}_i$, means that there is a physical link between G_i and G_j and it is assumed that in this case there is a communication link as well. Otherwise, $A_{ij} = 0$

and they are disconnected. Each subsystem is monitored by local sensors. Each sensor can only communicate with its neighbours and the transmission of information costs (unit) sampling time T. The topologies of physical and communication links among subsystems are the same.

6.3.2 An Optimal Fault Detection Scheme

Stack all subsystems to obtain the global model

$$x(k+1) = Ax(k) + w(k),\ \ y(k) = Cx(k) + v(k), \tag{6.41}$$

where

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, A = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{bmatrix}, \ C = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_N \end{bmatrix},$$

y, w and v have the same column structure with x, $x \in \mathcal{R}^n$ with $n = \sum_{i=1}^N n_i$ and $y \in \mathcal{R}^p$ with $p = \sum_{i=1}^N p_i$. Since assumption (6.40), $x(0)$ is uncorrelated with w and v,

$$x(0) \sim \mathcal{N}(\bar{x}_0, \Sigma),\ \ \text{cov}\left(\begin{bmatrix} w(k) \\ v(k) \end{bmatrix}, \begin{bmatrix} w^T(h) & v^T(h) \end{bmatrix} \right) = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \delta_{kh},$$

where $\bar{x}_0$ has same column structure with x and Q, R and Σ have the same block diagonal structure with C. Notice that the model (6.41) is the same as model (3.48), from (3.51)−(3.53), we have the optimal FD scheme for (6.41) as

$$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + L(k)r(k) \tag{6.42}$$

$$r(k) = y(k) - C\hat{x}(k|k-1) \tag{6.43}$$

$$\Sigma(k) = CP(k|k-1)C^T + R \tag{6.44}$$

$$L(k) = AP(k|k-1)C^T\Sigma(k)^{-1} \tag{6.45}$$

$$P(k+1|k) = AP(k|k-1)A^T + Q - L(k)\Sigma(k)L^T(k) \tag{6.46}$$

$$J(k) = r(k)\Sigma^{-1}(k)r^T(k),\ \ J_{th} = \chi^2_\alpha(p) \tag{6.47}$$

The procedures (6.42)-(6.46) is standard one-step prediction. Define $e(k) = x(k) - \hat{x}(k|k-1)$ as estimation error. As discussed in Chapter 3, the optimal FD is based on optimal state estimation by minimizing $P(k|k-1)$. Moreover, with fixed interval $[0, h]$, for $h \geqslant j \geqslant k \geqslant 0$, filtering and smoothing techniques [4] can be further applied to improve the estimation of

$x(k)$ by

$$\hat{x}(k|j) = \hat{x}(k|j-1) + K(j)C^T\Sigma^{-1}(j)r(j) \tag{6.48}$$

$$K(k) = P(k|k-1) \tag{6.49}$$

$$K(j+1) = K(j)(A - L(j)C)^T \tag{6.50}$$

$$P(k|j) = P(k|j-1) - K(j)C^T\Sigma^{-1}(j)CK^T(j) \tag{6.51}$$

with $\hat{x}(k|j-1)$, $P(k|k-1)$, $\Sigma(j)$, $r(j)$ delivered from (6.42)-(6.46). From (6.51), the variance of the estimation error is further reduced. Based on this observation, these techniques are involved in further study.

6.3.3 Problem Formulation

In Section 6.3.2, the optimal solution to detect fault for system (6.39) under a centralized manner is introduced. To compute $J(k)$ in (6.47), all $y_j(k)$, $j \in \mathcal{N}$, should be collected at node i. Since the transmission of information costs time, the computation of $J(k)$ can only be achieved at $(k + d_i)$ and thus needs MTTD d_i. Nevertheless, if a fault occurs in node i at time instant k and affects $y_i(k)$, it can be detected from time k onwards. Motivated by this observation, an iterative and distributed realization of optimal detection is considered and introduced in the rest of this chapter.

6.4 A Distributed Fault Detection Scheme for Dynamic Systems

In this section, we devote to propose a distributed FD scheme for model (6.39). For our purpose, notation $y_{\mathcal{N}_{i,s}}$, which has the same structure as (6.1), is defined as the collection of information on the $s-$layer neighbours of node i and

$$y_{\mathcal{N}_{i,s}}(k) = C_{\mathcal{N}_{i,s}}x(k) + v_{\mathcal{N}_{i,s}}(k).$$

Here, $v_{\mathcal{N}_{i,s}}$ has similar structure with $y_{\mathcal{N}_{i,s}}$ and with $\mathcal{N}_{i,s} = \{n_{i,s}^1, n_{i,s}^2, \cdots, n_{i,s}^{c_{i,s}}\}$, where $c_{i,s} = \mathrm{card}(\mathcal{N}_{i,s})$,

$$C_{\mathcal{N}_{i,s}} = \begin{bmatrix} C_{n_{i,s}^1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{n_{i,s}^{c_{i,s}}} \end{bmatrix}, \; R_{i,s} = \mathrm{var}(v_{\mathcal{N}_{i,s}}) = \begin{bmatrix} R_{n_{i,s}^1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{n_{i,s}^{c_{i,s}}} \end{bmatrix}.$$

Since the transmission of measurements from node $j \in \mathcal{N}_{i,\rho}$ to node i costs same time, we rearrange the measurement according to $d(i,j)$ and separate y into d_i parts as

$$\bar{y}_i = \begin{bmatrix} y^T_{\mathcal{N}_{i,0}} & y^T_{\mathcal{N}_{i,1}} & \cdots & y^T_{\mathcal{N}_{i,d_i}} \end{bmatrix}^T.$$

Note that $\bar{y}_i$ is a permutation of the rows of y in (6.41). Since the communication happens only between neighbours and costs unit sampling time, it spends ρ sampling time to transfer $y_{\mathcal{N}_{i,\rho}}$ to node i, which also means that node i can receive $y_{\mathcal{N}_{i,\rho}}(k-\rho)$ at time instant k. The optimal centralized method in Section 6.3.2 is achieved by the minimum variance estimation. Intuitively speaking, if more measurements are used for FD in node i, the uncertainty will be reduced and the accuracy of the FD will increase. However, the MTTD will also increase. Similar to the static case, in this section, we first set a criterion to decide how much information is needed for FD in node i, then achieve the distributed FD based on the selected information.

6.4.1 Node Selection

The selection of node is also based on $d(i,j)$ as in Section 6.2.1, it means that if the measurements inside $m-$layer can not satisfy our condition, we continuously select more measurements inside $(m+1)-$layer to check the condition until it is fulfilled. First, we quantify the improvement of more measurements for FD in G_i. Assume that the measurements from $0-$layer to $m-$layer neighbours are used to detect faults in G_i. Define the collection of measurements inside $m-$layer as $\bar{y}_{i,m} = \begin{bmatrix} y_{\mathcal{N}_{i,0}} & y_{\mathcal{N}_{i,1}} & \cdots & y_{\mathcal{N}_{i,m}} \end{bmatrix}$ with $\bar{y}_{i,m} \in \mathcal{R}^{p_{i,m}}$ and

$$\bar{y}_{i,m}(k) = \begin{bmatrix} y_{\mathcal{N}_{i,0}}(k) & \cdots & y_{\mathcal{N}_{i,m}}(k) \end{bmatrix} = \bar{C}_{i,m}x(k) + \bar{v}_{i,m}(k), \tag{6.52}$$

where $\bar{v}_{i,m}$ has similar structure with $\bar{y}_{i,m}$,

$$\bar{C}_{i,m} = \begin{bmatrix} C_{\mathcal{N}_{i,0}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{\mathcal{N}_{i,m}} \end{bmatrix}, \ \bar{R}_{i,m} = \mathrm{var}(v_{\mathcal{N}_{i,s}}) = \begin{bmatrix} R_{i,0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{i,m} \end{bmatrix}.$$

The model for G_i with measurements inside $m-$ layer is

$$x(k+1) = Ax(k) + w(k), \ \bar{y}_{i,m}(k) = \bar{C}_{i,m}x(k) + \bar{v}_{i,m}(k). \tag{6.53}$$

Set $O_{i,m}$ as the observable subspace of (6.53), $\dim(O_{i,m}) = n_{i,m}$, the orthonormal basis of $O_{i,m}$ as

$$b_{i,m} = \begin{bmatrix} b_1 \\ \vdots \\ b_{n_{i,m}} \end{bmatrix} \in \mathcal{R}^{n_{i,m} \times n}, \ \mathrm{rank}(b_{i,m}) = n_{i,m}, \tag{6.54}$$

and its orthogonal complement

$$\bar{b}_{i,m} = \begin{bmatrix} b_{n_{i,m}+1} \\ \vdots \\ b_n \end{bmatrix} \in \mathcal{R}^{(n-n_{i,m})\times n}, \ \text{rank}(\bar{b}_{i,m}) = n - n_{i,m}. \tag{6.55}$$

Set $T_{i,m} = \begin{bmatrix} b_{i,m}^T & \bar{b}_{i,m}^T \end{bmatrix}^T$ as coordinate transformation matrix for (6.53) according to pair $(\bar{C}_{i,m}, A)$ such that the linear transformation $\xi_{i,m} = T_{i,m}x = \begin{bmatrix} (\xi_{i,m}^o)^T & (\xi_{i,m}^u)^T \end{bmatrix}^T$, where $\xi_{i,m}^o \in \mathcal{R}^{n_{i,m}}$ and $\xi_{i,m}^u \in \mathcal{R}^{n-n_{i,m}}$ stand for the observable and unobservable part respectively, transforms the original state space representation (6.53) into the following observability staircase form [129]

$$\begin{aligned} \xi_{i,m}(k+1) &= \begin{bmatrix} A_{i,m} & 0 \\ * & \bar{A}_{i,m} \end{bmatrix} \xi_{i,m}(k) + \begin{bmatrix} w_{i,m}(k) \\ \bar{w}_{i,m}(k) \end{bmatrix}. \\ \bar{y}_{i,m}(k) &= \begin{bmatrix} C_{i,m} & 0 \end{bmatrix} \xi_{i,m}(k) + \bar{v}_{i,m}(k) \end{aligned} \tag{6.56}$$

Apply Kalman filter to estimate observable part $\xi_{i,m}^o$,

$$\hat{\xi}_{i,m}^o(k+1|k) = A_{i,m}\hat{\xi}_{i,m}^o(k|k-1) + L_{i,m}(k)r_{i,m}(k) \tag{6.57}$$

$$r_{i,m}(k) = \bar{y}_{i,m}(k) - C_{i,m}\hat{\xi}_{i,m}^o(k|k-1), \tag{6.58}$$

$$\Sigma_{i,m}(k) = C_{i,m}P_{i,m}(k|k-1)C_{i,m}^T + \bar{R}_{i,m}, \tag{6.59}$$

$$L_{i,m}(k) = A_{i,m}P_{i,m}(k|k-1)C_{i,m}^T\Sigma_{i,m}^{-1}(k), \tag{6.60}$$

$$P_{i,m}(k+1|k) = A_{i,m}P_{i,m}(k|k-1)A_{i,m}^T + Q_{i,m} - L_{i,m}(k)\Sigma_{i,m}(k)L_{i,m}^T(k), \tag{6.61}$$

where $Q_{i,m} = \text{var}(w_{i,m}) = b_{i,m}Qb_{i,m}^T$. Since our aim is to estimate y_i, variable $\xi_{i,0}^o$, which is the observable part of (C_i, A), need to be recovered from $\xi_{i,m}^o$. It is directly achieved by linear transformation

$$\xi_{i,0}^o(k) = G_{i,m}\xi_{i,m}^o(k), \ G_{i,m} = \begin{cases} I_{i,0}, & O_{i,0} = O_{i,m} \\ \begin{bmatrix} I_{i,0} & 0 \end{bmatrix}, & O_{i,0} \neq O_{i,m} \end{cases}, \tag{6.62}$$

where $I_{i,0}$ is the identity matrix and has the same dimension with $A_{i,0}$. And the estimation variance is

$$\text{var}(\xi_{i,0}^o(k) - \hat{\xi}_{i,0}^o(k)) = G_{i,m}P_{i,m}(k|k-1)G_{i,m}^T.$$

Theorem 6.2. *Suppose $P_{i,m}$ is the steady-state solution of (6.61). If more layers are selected for estimation, we have*

$$x(k+1) = Ax(k) + w(k), \ \bar{y}_{i,g}(k) = \bar{C}_{i,g}x(k) + \bar{v}_{i,g}(k), \ g > m, \tag{6.63}$$

then estimation variance of $\xi_{i,0}^o$ is reduced with

$$G_{i,g}P_{i,g}G_{i,g}^T \leqslant G_{i,m}P_{i,m}G_{i,m}^T.$$

Proof. Set $O_{i,g}$ as the observable subspace according to pair $(\bar{C}_{i,g}, A)$. It holds that $\dim(O_{i,g}) = \dim(O_{i,m})$ or $\dim(O_{i,g}) > \dim(O_{i,m})$. For $\dim(O_{i,g}) = \dim(O_{i,m})$,we have $T_{i,g} = T_{i,m}$ and the corresponding staircase form is:

$$\begin{bmatrix} \xi^o_{i,g}(k+1) \\ \xi^u_{i,g}(k+1) \end{bmatrix} = \begin{bmatrix} A_{i,m} & 0 \\ * & \bar{A}_{i,m} \end{bmatrix} \begin{bmatrix} \xi^o_{i,g}(k) \\ \xi^u_{i,g}(k) \end{bmatrix} + \begin{bmatrix} w_{i,m}(k) \\ \bar{w}_{i,m}(k) \end{bmatrix}$$
$$\bar{y}_{i,g}(k) = \begin{bmatrix} C_{i,m} & 0 \\ C_{i,m,g} & 0 \end{bmatrix} \begin{bmatrix} \xi^o_{i,g}(k) \\ \xi^u_{i,g}(k) \end{bmatrix} + \bar{v}_{i,g}(k),$$

where $C_{i,g} = \begin{bmatrix} C^T_{i,m} & C^T_{i,m,g} \end{bmatrix}^T$,

$$\operatorname{var}(\bar{v}_{i,g}) = \bar{R}_{i,g} = \begin{bmatrix} \bar{R}_{i,m} & 0 \\ 0 & \bar{R}_{i,m,g} \end{bmatrix}, \ \bar{R}_{i,m,g} = \begin{bmatrix} R_{i,m+1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{i,g} \end{bmatrix}.$$

Apply Kalman filter to estimate observable part $\xi^o_{i,g}$,

$$\hat{\xi}^o_{i,g}(k+1|k) = A_{i,m}\hat{\xi}^o_{i,g}(k|k-1) + L_{i,g}(k)r_{i,g}(k), \tag{6.64}$$
$$r_{i,g}(k) = \bar{y}_{i,g}(k) - C_{i,g}\hat{\xi}^o_{i,g}(k|k-1), \tag{6.65}$$
$$\Sigma_{i,g}(k) = C_{i,g}P_{i,g}(k|k-1)C^T_{i,g} + \bar{R}_{i,g}, \tag{6.66}$$
$$L_{i,g}(k) = A_{i,m}P_{i,g}(k|k-1)C^T_{i,g}\Sigma^{-1}_{i,g}(k), \tag{6.67}$$
$$P_{i,g}(k+1|k) = A_{i,m}P_{i,g}(k|k-1)A^T_{i,m} + Q_{i,m} - L_{i,g}(k)\Sigma_{i,g}(k)L^T_{i,g}(k). \tag{6.68}$$

Modify (6.68) as

$$P_{i,g}(k+1|k) = A_{i,m}P_{i,g}(k|k-1)A^T_{i,m} + Q_{i,m} - H_1 - H_2H_3H_2^T, \tag{6.69}$$
$$H_2 = A_{i,m}P_{i,g}(k|k-1)C^T_{i,m}H_3^{-1}, \ H_3 = C_{i,m}P_{i,g}(k|k-1)C^T_{i,m} + \bar{R}_{i,m},$$
$$H_1 = H_4(\Sigma_2 - \Sigma_{21}\Sigma_1^{-1}\Sigma_{12})^{-1}H_4^T, \ H_4 = A_{i,m}P_{i,g}(k|k-1)(C^T_{i,m,g} - C^T_{i,m}\Sigma_1^{-1}\Sigma_{12}),$$
$$\Sigma_1 = C_{i,m}P_{i,g}(k|k-1)C^T_{i,m} + \bar{R}_{i,m}, \ \Sigma_2 = C_{i,m,g}P_{i,g}(k|k-1)C^T_{i,m,g} + \bar{R}_{i,m,g},$$
$$\Sigma_{12} = C_{i,m}P_{i,g}(k|k-1)C^T_{i,m,g}, \ \Sigma_{21} = \Sigma^T_{12}.$$

Hence $\Sigma_{i,g}(k) > 0$ and Schur complement, it holds that

$$H_1 \geqslant 0. \tag{6.70}$$

With initial value $P_{i,g}(0|-1) = P_{i,m}(0|-1)$, combine (6.61), (6.69) and (6.70), we have

$$P_{i,g}(1) - P_{i,m}(1) = -H_1 \leqslant 0 \ \Rightarrow P_{i,g}(1) \leqslant P_{i,m}(1). \tag{6.71}$$

From [7], we have

$$P_{i,g}(k) \leqslant P_{i,m}(k) \Rightarrow P_{i,g}(k+1) \leqslant P_{i,m}(k+1) \tag{6.72}$$

and

$$P_{i,g} = \lim_{k\to\infty} P_{i,g}(k) \leqslant P_{i,m} = \lim_{k\to\infty} P_{i,m}(k). \tag{6.73}$$

Hence $b_{i,g} = b_{i,m}$ and (6.73), we have $G_{i,g} = G_{i,m}$ and

$$G_{i,g}P_{i,g}G_{i,g}^T \leqslant G_{i,m}P_{i,m}G_{i,m}^T. \tag{6.74}$$

When $\dim(O_{i,g}) > \dim(O_{i,m})$, set $\dim(O_{i,g}) = n_{i,g}$, the orthonormal basis of $O_{i,g}$ as $b_{i,g} = \begin{bmatrix} b_{i,m}^T & b_{i,m,g}^T \end{bmatrix}^T$, where $b_{i,m,g}$ stands for the orthonormal basis of new observable subspace, $\bar{b}_{i,g}$ as the orthogonal complement of $b_{i,g}$ and transformation matrix $T_{i,g} = \begin{bmatrix} b_{i,g}^T & \bar{b}_{i,g}^T \end{bmatrix}^T$ to obtain the observability staircase form of (6.63). Apply state transformation $\xi_{i,g} = T_{i,g}x$ to obtain

$$\begin{aligned} \begin{bmatrix} \xi_{i,g}^o(k+1) \\ \xi_{i,g}^u(k+1) \end{bmatrix} &= \begin{bmatrix} A_{i,g} & 0 \\ * & \bar{A}_{i,g} \end{bmatrix} \begin{bmatrix} \xi_{i,g}^o(k) \\ \xi_{i,g}^u(k) \end{bmatrix} + \begin{bmatrix} w_{i,g}(k) \\ \bar{w}_{i,g}(k) \end{bmatrix} \\ \bar{y}_{i,g}(k) &= \begin{bmatrix} C_{i,g} & 0 \end{bmatrix} \begin{bmatrix} \xi_{i,g}^o(k) \\ \xi_{i,g}^u(k) \end{bmatrix} + \bar{v}_{i,g}(k), \end{aligned} \tag{6.75}$$

where $\xi_{i,g}^o = \begin{bmatrix} (\xi_{i,m}^o)^T & (\xi_{i,m,g}^o)^T \end{bmatrix}^T$, $\xi_{i,m,g}^o$ stands for the new observable state,

$$A_{i,g} = \begin{bmatrix} A_{i,m} & 0 \\ Z_{i,m,g} & A_{i,m,g} \end{bmatrix}, \; C_{i,g} = \begin{bmatrix} C_{i,m} & 0 \\ X_{i,m,g} & C_{i,m,g} \end{bmatrix}, \; w_{i,g} = \begin{bmatrix} w_{i,m} \\ b_{i,m,g}w \end{bmatrix}$$
$$\operatorname{var}(w_{i,g}(k)) = Q_{i,g} = \begin{bmatrix} Q_{i,m} & b_{i,m}Qb_{i,m,g}^T \\ b_{i,m,g}Qb_{i,m}^T & b_{i,m,g}Qb_{i,m,g}^T \end{bmatrix}$$

Use Kalman filter to estimate observable part $\xi_{i,g}^o$,

$$\hat{\xi}_{i,g}^o(k+1|k) = A_{i,g}\hat{\xi}_{i,g}^o(k|k-1) + L_{i,g}(k)r_{i,g}(k), \tag{6.76}$$
$$r_{i,g}(k) = \bar{y}_{i,g}(k) - C_{i,g}\hat{\xi}_{i,g}^o(k), \tag{6.77}$$
$$\Sigma_{i,g}(k) = C_{i,g}P_{i,g}(k|k-1)C_{i,g}^T + \bar{R}_{i,g} \tag{6.78}$$
$$L_{i,g}(k) = A_{i,g}P_{i,g}(k|k-1)C_{i,g}^T\Sigma_{i,g}^{-1}(k) \tag{6.79}$$
$$P_{i,g}(k+1|k) = A_{i,g}P_{i,g}(k|k-1)A_{i,g}^T + Q_{i,g} - L_{i,g}(k)\Sigma_{i,g}(k)L_{i,g}^T(k), \tag{6.80}$$

For further analysis, separate

$$P_{i,g}(k|k-1) = \begin{bmatrix} P_{i,m}^g(k|k-1) & S_{i,m,g}(k|k-1) \\ S_{i,m,g}^T(k|k-1) & P_{i,m,g}(k|k-1) \end{bmatrix}$$

where $P^g_{i,m}$, $P_{i,m,g}$ and $S_{i,m,g}$ stand for the estimation variance of $\xi^o_{i,m}$, $\xi^o_{i,m,g}$ and covariance matrix between them, respectively. And the iteration of $P^g_{i,m}(k|k-1)$ in (6.80) is

$$\begin{aligned}
&P^g_{i,m}(k+1|k) = A_{i,m}P^g_{i,m}(k|k-1)A^T_{i,m} + Q_{i,m} - H_1 - H_2H_3H_2^T,\\
&H_2 = A_{i,m}P^g_{i,m}(k)C^T_{i,m}H_3^{-1},\ H_3 = C_{i,m}P^g_{i,m}(k|k-1)C^T_{i,m} + \bar{R}_{i,m},\\
&H_1 = H_4(\Sigma_2 - \Sigma_{21}\Sigma_1^{-1}\Sigma_{12})^{-1}H_4^T,\ \Sigma_1 = C_{i,m}P^g_{i,m}(k|k-1)C^T_{i,m} + \bar{R}_{i,m},\\
&H_4 = A_{i,m}(P^g_{i,m}(k)X^T_{i,m,g} + S_{i,m,g}(K)C^T_{i,m,g} - P^g_{i,m}(k)C^T_{i,m}\Sigma_1^{-1}\Sigma_{12}),\\
&\Sigma_2 = X_{i,m,g}P^g_{i,m}(k|k-1)X^T_{i,m,g} + C_{i,m,g}P_{i,m,g}(k|k-1)C^T_{i,m,g} + \bar{R}_{i,m,g},\\
&\Sigma_{12} = C_{i,m}P^g_{i,m}(k|k-1)X^T_{i,m,g} + C_{i,m}S_{i,m,g}(k|k-1)C^T_{i,m,g},\ \Sigma_{21} = \Sigma_{12}^T.
\end{aligned} \tag{6.81}$$

With the similar steps from (6.70) to (6.73), it holds that

$$P^g_{i,m} \leqslant P_{i,m}. \tag{6.82}$$

Since $b_{i,g} = \begin{bmatrix} b^T_{i,m} & b^T_{i,m,g}\end{bmatrix}^T$, we have $G_{i,g} = \begin{bmatrix} G_{i,m} & 0\end{bmatrix}$ and

$$G_{i,g}P_{i,g}G^T_{i,g} = G_{i,m}P^g_{i,m}G^T_{i,m} \leqslant G_{i,m}P_{i,m}G^T_{i,m}. \tag{6.83}$$

Combine (6.74) and (6.83), Theorem 6.2 is proved. □

Theorem 6.2 presents that if we use more measurements to do estimation, the stronger influence of uncertainty on $\xi^o_{i,0}$ can be reduced. Thus the estimation becomes more accurate and the detection performance is also improved. For selecting node for G_i, since $\hat{\xi}^o_{i,0}$ is the minimal observable subspace to recover y_i, set η_{acc} as an acceptable level for the accuracy of its estimation and find ρ_i such that

$$\begin{aligned}
\rho_i = \arg\min_{m\in\mathbb{Z}} \quad & m\\
\text{s.t.} \quad & tr(G_{i,m}P_{i,m}G^T_{i,m}) \leqslant \eta_{acc}
\end{aligned}. \tag{6.84}$$

6.4.2 Distributed Estimation

Assume ρ_i is the solution of (6.84), it means that the measurements inside ρ_i-layer are used for FD in G_i. With the assumption of transmission time, at time instant k, node i receives new measurement data

$$y_{i,\rho_i}(k) = \begin{bmatrix} y^T_{\mathcal{N}_{i,0}}(k) & \cdots & y^T_{\mathcal{N}_{i,\rho_i}}(k-\rho_i)\end{bmatrix}^T. \tag{6.85}$$

Notice that (6.85) is modified from (6.52) when considering transmission time. If we apply (6.57)–(6.61) for estimation and FD, at time instant k, $\bar{y}_{i,\rho_i}(k-\rho_i)$ is obtained and used for estimate $\xi^o_{i,\rho_i}(k-\rho_i+1)$. But data

$$y_{\mathcal{N}_{i,j}}(k-j),\ j = 0, \cdots, \rho_i - 1,$$

which are also received within k, are not involved in the above procedure, and can be used to improve the estimation result. In the rest part of this section, we present a distributed algorithm that makes use of all data inside time k. For simplicity, in the subsequent part of this paper, the symbol $(k|k-1)$, which indicates the one-step prediction process, is shortened to (k). We first introduce the following theorem and lemmas.

Theorem 6.3. *Give $y_{i,\rho_i}(k)$ in (6.85), initial value $\hat{\xi}^o_{i,\rho_i}(k-\rho_i)$ and $P_{i,\rho_i}(k-\rho_i)$ at time instant k. For $h=\rho_i,\cdots,1$,*

$$\hat{\xi}^o_{i,h}(k-h+1) = A_{i,h}\hat{\xi}^o_{i,h}(k-h) + L_{i,h}(k-h)r_{i,h}(k-h), \tag{6.86}$$

$$P_{i,h}(k-h+1) = A_{i,h}P_{i,h}(k-h)A^T_{i,h} + Q_{i,h} - L_{i,h}(k-h)\Sigma_{i,h}(k-h)L^T_{i,h}(k-h), \tag{6.87}$$

$$\hat{\xi}^o_{i,h-1}(k-h+1) = N_{i,h}\hat{\xi}^o_{i,h}(k-h+1), \tag{6.88}$$

$$P_{i,h-1}(k-h+1) = N_{i,h}P_{i,h}(k-h+1)N^T_{i,h}, \tag{6.89}$$

$$N_{i,h} = \begin{cases} I_{i,h-1}, & O_{i,h} = O_{i,h-1} \\ \begin{bmatrix} I_{i,h-1} & 0 \end{bmatrix}, & O_{i,h} \neq O_{i,h-1} \end{cases}, \tag{6.90}$$

for $h=\rho_i,\cdots,0$,

$$r_{i,h}(k-h) = \bar{y}_{i,h}(k-h) - C_{i,h}\hat{\xi}^o_{i,h}(k-h), \tag{6.91}$$

$$\Sigma_{i,h}(k-h) = C_{i,h}P_{i,h}(k-h)C^T_{i,h} + \bar{R}_{i,h}, \tag{6.92}$$

$$L_{i,h}(k-h) = A_{i,h}P_{i,h}(k-h)C^T_{i,h}\Sigma^{-1}_{i,h}(k-h), \tag{6.93}$$

where $I_{i,h-1}$ is the identity matrix and has the same dimension with $A_{i,h-1}$, $\hat{\xi}^o_{i,0}(k)$ delivers the minimal variance estimation for $\xi^o_{i,0}(k)$.

Proof. Select $y_{\mathcal{N}_{i,\rho_i}}(k-\rho_i)$ from $y_{i,\rho_i}(k)$ and obtain

$$\bar{y}_{i,\rho_i}(k-\rho_i) = \begin{bmatrix} y_{\mathcal{N}_{i,0}}(k-\rho_i) \\ \vdots \\ y_{\mathcal{N}_{i,\rho_i}}(k-\rho_i) \end{bmatrix},$$

where $y_{\mathcal{N}_{i,0}}(k-\rho_i)$, $\ldots$, $y_{\mathcal{N}_{i,\rho_i-1}}(k-\rho_i+1)$ are from the previous sampling interval. With $\bar{y}_{i,\rho_i}(k-\rho_i)$, $\hat{\xi}^o_{i,\rho_i}(k-\rho_i)$ and $P_{i,\rho_i}(k-\rho_i)$, the minimal variance one-step prediction $\hat{\xi}^o_{i,\rho_i}(k-\rho_i+1)$ and variance matrix $P_{i,\rho_i}(k-\rho_i+1)$ can be obtained from

$$r_{i,\rho_i}(k-\rho_i) = \bar{y}_{i,\rho_i}(k-\rho_i) - C_{i,\rho_i}\hat{\xi}^o_{i,\rho_i}(k-\rho_i),$$

$$\Sigma_{i,\rho_i}(k-\rho_i) = C_{i,\rho_i}P_{i,\rho_i}(k-\rho_i)C^T_{i,\rho_i} + \bar{R}_{i,\rho_i},$$

$$L_{i,\rho_i}(k-\rho_i) = A_{i,\rho_i}P_{i,\rho_i}(k-\rho_i)C^T_{i,\rho_i}\Sigma^{-1}_{i,\rho_i}(k-\rho_i),$$

$$\hat{\xi}^o_{i,\rho_i}(k-\rho_i+1) = A_{i,\rho_i}\hat{\xi}^o_{i,\rho_i}(k-\rho_i) + L_{i,\rho_i}(k-\rho_i)r_{i,\rho_i}(k-\rho_i),$$

$$P_{i,\rho_i}(k-\rho_i+1) = A_{i,\rho_i}P_{i,\rho_i}(k-\rho_i)A^T_{i,\rho_i} + Q_{i,\rho_i} - L_{i,\rho_i}(k-\rho_i)\Sigma_{i,\rho_i}(k-\rho_i)L^T_{i,\rho_i}(k-\rho_i),$$

which are standard Kalman filter procedures. Similar with (6.62), it holds that

$$\hat{\xi}^o_{i,\rho_i-1}(k-\rho_i+1) = N_{i,\rho_i}\hat{\xi}^o_{i,\rho_i}(k-\rho_i+1), P_{i,\rho_i-1}(k-\rho_i+1) = N_{i,\rho_i}P_{i,\rho_i}(k-\rho_i+1)N^T_{i,\rho_i}$$

with $N_{i,\rho_i} = \begin{cases} I_{i,\rho_i-1}, & O_{i,\rho_i} = O_{i,\rho_i-1} \\ \begin{bmatrix} I_{i,\rho_i-1} & 0 \end{bmatrix}, & O_{i,\rho_i} \neq O_{i,\rho_i-1} \end{cases}$. Applying the same procedure to predict the observable subspace from $\hat{\xi}^o i, \rho i - 1(k-\rho_i+2)$ to $\hat{\xi}^o i, 1(k+1)$ yields the identical steps as presented in Theorem 6.3. According to Theorem 6.2, for $h = \rho_i, \cdots, 1$, the optimal estimation with minimal variance is $\hat{\xi}^o_{i,h-1}(k-h+1)$ when provided with $\bar{y}_{i,h}(k-h)$. By mathematical induction, it is evident that $\hat{\xi}^o_{i,0}(k)$ represents the minimal variance estimation, given $\hat{\xi}^o_{i,\rho_i}(k-\rho_i)$, $P_{i,\rho_i}(k-\rho_i)$ and the dataset $y_{i,\rho_i}(k)$. □

Lemma 6.3. [64] *Given the one-step prediction in Theorem 6.3 and set $e_{i,h} = \xi^o_{i,h} - \hat{\xi}^o_{i,h}$ as the estimation error, the signals $r_{i,\rho_i}(k)$ are innovations and it holds that*

$$\operatorname{cov}\left(r_{i,\rho_i}(j), r_{i,\rho_i}(p)\right) = \Sigma_{i,\rho_i}(j)\delta_{jp} \tag{6.94}$$

$$e_{i,h}(k-h+1) = \bar{A}_{i,h}(k-h)e_{i,h}(k-h) + w_{i,h}(k-h) - L_{i,h}(k-h)\bar{v}_{i,h}(k-h) \tag{6.95}$$

$$\bar{A}_{i,h}(k-h) = (A_{i,h} - L_{i,h}(k-h)C_{i,h})$$

$$e_{i,h-1}(k-h+1) = N_{i,h}e_{i,h}(k-h+1) \tag{6.96}$$

$$r_{i,h}(k-h) = C_{i,h}e_{i,h}(k-h) + \bar{v}_{i,h}(k-h) \tag{6.97}$$

$$e_{i,h}(k-h+1) \sim (0, P_{i,h}(k-h+1)) \tag{6.98}$$

$$e_{i,h-1}(k-h+1) \sim (0, N_{i,h}P_{i,h}(k-h+1)N^T_{i,h}) \tag{6.99}$$

$$r_{i,h}(k-h) \sim (0, \Sigma_{i,h}(k-h)) \tag{6.100}$$

Lemma 6.4. *Given the one-step prediction in Theorem 6.3, at time instant k, for $j, p = 0, \cdots, \rho_i$ and $h = 0, \cdots, k-\rho_i-1$, it holds that*

$$\operatorname{cov}(r_{i,j}(k-j), r_{i,p}(k-p)) = \Sigma_{i,j}(k-j)\delta_{jp} \tag{6.101}$$

$$\operatorname{cov}(r_{i,j}(k-j), r_{i,p_i}(h)) = 0. \tag{6.102}$$

Proof. For $j \neq p$, without loss of general, assume $j < p$. From (6.95) and (6.96),

$$\begin{aligned} r_{i,j}(k-j) &= C_{i,j}e_{i,j}(k-j) + \bar{v}_{i,j}(k-j) \\ &= C_{i,j}\Phi^p_{i,j}e_{i,p}(k-p) + f^p_{i,j} - C_{i,j}\Phi^{p-1}_{i,j}N_{i,p}L_{i,p}\bar{v}_{i,p}(k-p), \end{aligned} \tag{6.103}$$

where

$$\Phi^p_{i,j} = \prod_{q=j+1}^{p} N_{i,q}(A_{i,q} - L_{i,q}(k-q)C_{i,q}) \tag{6.104}$$

and $f_{i,j}^p$ is the linear combination of noise signals $w_{i,p}(k-p), \cdots, w_{i,j+1}(k-j-1)$ and $\bar{v}_{i,p-1}(k-p+1), \cdots, \bar{v}_{i,j}(k-j)$ with corresponding weighting matrices and is uncorrelated with $r_{i,p}(k-p)$. Based on decomposition (6.103), it holds that

$$\begin{aligned}
&\operatorname{cov}(r_{i,j}(k-j), r_{i,p}(k-p)) \qquad (6.105)\\
=&\operatorname{cov}(C_{i,j}\Phi_{i,j}^p e_{i,p}(k-p) - C_{i,j}\Phi_{i,j}^{p-1} N_p L_{i,p}\bar{v}_{i,p}(k-p) + f_{i,j}^p, C_{i,p}e_{i,p}(k-p) + \bar{v}_{i,p}(k-p))\\
=&C_{i,j}\Phi_{i,j}^p P_{i,p}(k-p)C_{i,p}^T - C_{i,j}\Phi_{i,j}^{p-1} N_p L_{i,p}\bar{R}_{i,p} = 0.
\end{aligned}$$

Combining (6.100) and (6.105), (6.101) is proved. Since equation (6.86) is standard one-step prediction, it holds that [64]

$$\operatorname{cov}(e_{i,\rho_i}(k-\rho_i), r_{i,\rho_i}(h)) = 0. \qquad (6.106)$$

Based on (6.103) and (6.106), we have

$$\begin{aligned}
&\operatorname{cov}(r_{i,j}(k-j), r_{i,\rho_i}(h))\\
=&\operatorname{cov}(C_{i,j}\Phi_{i,j}^{\rho_i} e_{i,\rho_i}(k-\rho_i) + f_{i,j}^{\rho_i} - C_{i,j}\Phi_{i,j}^{\rho_i -} N_{\rho_i}\bar{v}_{i,\rho_i}(k-\rho_i), r_{i,\rho_i}(h)) = 0.
\end{aligned}$$

Thus (6.102) is proved and the proof is done. □

Theorem 6.3 provides a distributed realization of one-step prediction to estimate observable subspace. After computation, we obtain the one-step predictions

$$\hat{\xi}_{i,0}^o(k),\ \hat{\xi}_{i,1}^o(k-1),\ \cdots,\ \hat{\xi}_{i,\rho_i-1}^o(k-\rho_i+1), \qquad (6.107)$$

and also the sequence of residuals

$$r_{i,0}(k),\ r_{i,1}(k-1),\ \cdots,\ r_{i,\rho_i}(k-\rho_i), \qquad (6.108)$$

which are innovations and mutually uncorrelated according to Lemma 6.4 and [64]. Also notice that for $h = 0, \cdots, \rho_i - 1$, signal $\hat{\xi}_{i,h}^o(k-h)$ is predicted based on

$$r_{i,h+1}(k-h-1),\ \cdots,\ r_{i,\rho_i-1}(k-\rho_i+1),\ r_{i,\rho_i}(k-\rho_i)$$

and other residuals

$$r_{i,0}(k),\ \cdots,\ r_{i,1}(k-1),\ r_{i,h}(k-h),$$

which are generated after the computation of $\hat{\xi}_{i,h}^o(k-h)$, are not involved in the prediction of $\xi_{i,h}^o(k-h)$ and thus can be further applied to improve the estimation result of $\xi_{i,h}^o(k-h)$ by meanings of filtering and smoothing.

Theorem 6.4. *[64] Given (6.107) and (6.108), for $h = 0, \cdots, \rho_i$,*

$$
\begin{aligned}
\hat{\xi}^{o,fs}_{i,h}(k-h) &= \hat{\xi}^{o}_{i,h}(k-h) + \sum_{z=0}^{h} \bar{L}_h(z) r_{i,z}(k-z) && (6.109)\\
\bar{L}_h(h) &= P_{i,h}(k-h) C^T_{i,h} \Sigma^{-1}_{i,h}(k-h), && (6.110)\\
\bar{L}_h(z) &= P_{i,h}(k-h) (\Phi^h_{i,z})^T C^T_{i,z} \Sigma^{-1}_{i,z}(k-z), 0 \leqslant z \leqslant h-1 &&
\end{aligned}
$$

are the optimal filtering and smoothing steps to improve the estimation result by using all residuals generated within time index k. Define $e^{o,fs}_{i,h}(k-h) = \xi^{o}_{i,h}(k-h) - \hat{\xi}^{o,fs}_{i,h}(k-h)$, it holds that $e^{o,fs}_{i,h}(k-h) \sim (0, P^{fs}_{i,h}(k-h))$ and

$$
P^{fs}_{i,h}(k-h) = P_{i,h}(k-h) - \textstyle\sum_{z=0}^{h} \bar{L}_h(z) \Sigma_{i,z}(k-z) \bar{L}^T_h(z). \tag{6.111}
$$

Notice that the estimation of $\xi^{o}_{i,h}(k-h)$ in Theorem 6.4 is improved by the innovations $r_i(k), \cdots, r_{i,h}(k-h)$, which carries new information from present and future measurement. Also from (6.111), it is obvious that $P^{fs}_{i,h}(k-h) \leqslant P_{i,h}(k-h)$, so the accuracy of our estimation is improved by including filtering and smoothing.

6.4.3 Distributed Fault Detection

Since it needs time to achieve the filtering and smoothing steps, the estimation results delivered by one-step prediction is first used for FD. Also, with a more accurate subspace estimation, the results from filtering and smoothing is then applied to detect faults. For this purpose, build residual signals between y_i and its estimations and for $h = 0, \cdots, \rho_i$, compute residuals and their variance matrices

$$
\begin{aligned}
\bar{r}_{i,h}(k-h) =& S_{i,h}(\bar{y}_{i,h}(k-h) - C_{i,h}\hat{\xi}^{o}_{i,h}(k-h)), && (6.112)\\
\bar{\Sigma}_{i,h}(k-h) =& S_{i,h}\Sigma_{i,h}(k-h)S^T_{i,h}, && (6.113)\\
\bar{r}^{fs}_{i,h}(k-h) =& S_{i,h}(\bar{y}_{i,h}(k-h) - C_{i,h}\hat{\xi}^{o,fs}_{i,h}(k-h)), && (6.114)\\
\bar{\Sigma}^{fs}_{i,0}(k) =& \bar{R}_{i,0}\Sigma^{-1}_{i,0}(k)\bar{R}_{i,0}, && (6.115)\\
\bar{\Sigma}^{fs}_{i,j}(k-j) =& S_{i,h}\bar{R}_{i,j}\Sigma^{-1}_{i,j}(k-j)\bar{R}_{i,j}S^T_{i,h} && (6.116)\\
& + \sum_{z=0}^{j-1} S_{i,j}C_{i,j}\bar{L}_j(z)\Sigma_{i,z}(k-z)\bar{L}_j(z)^T C^T_{i,j}S^T_{i,j}, 1 \leqslant j \leqslant \rho_i, , &&
\end{aligned}
$$

where $S_{i,h} = \begin{bmatrix} I_{p_i} & 0 \end{bmatrix}$ and I_{p_i} is identity matrix with dimension p_i. From (6.52), y_i locates at first p_i rows of $\bar{y}_i(k-h)$, so $S_{i,h}$ is applied as selecting matrix to build the residual for y_i. Since $\bar{r}_{i,h}(k-h) \sim \mathcal{N}(0, \bar{\Sigma}_{i,h}(k-h))$, $\bar{r}^{fs}_{i,h}(k-h) \sim \mathcal{N}(0, \bar{\Sigma}^{fs}_{i,h}(k-h))$ and according to

(6.47), set evaluation function and threshold as

$$\begin{aligned}J_{i,h}(k-h) &= \bar{r}_{i,h}^{T}(k-h)\bar{\Sigma}_{i,h}^{-1}(k-h)\bar{r}_{i,h}(k-h),\\ J_{i,h}^{fs}(k-h) &= (\bar{r}_{i,h}^{fs}(k-h))^{T}(\bar{\Sigma}_{i,h}^{fs}(k-h))^{-1}\bar{r}_{i,h}^{fs}(k-h),\\ J_{i,th} &= \chi_{\alpha}^{2}(p_i)\end{aligned} \tag{6.117}$$

with given α, then apply decision logic (3.1) to do FD.

The core part of the proposed distributed method is that we use prediction, filtering and smoothing procedures to improve the accuracy of estimation by means of reducing the variance matrix of the estimation error. The method makes full use of data, which node i receives within time instant k, to do optimal estimation. As shown in Figure 6.3, we first run one-step prediction based on the measured data, which node i received within time instant k, to obtain the estimation and also the innovation sequences. Then the innovation sequences are applied to improve the estimation delivered from one-step prediction by filtering and smoothing. The overall algorithm is summarised in Algorithm 6.2.

Algorithm 6.2. *Distributed FD for dynamic systems*

```
Offline Computation
For $i = 1, \cdots, N$
Step 1   Calculate $d_i$ at node $i$
Step 2   Calculate $b_{i,m}$, $A_{i,m}$ and $C_{i,m}$ using (6.52)−(6.56)
Step 3   Calculate $P_{i,m}$ as steady state for (6.61)
Step 4   Set $\eta_{acc}$ and select $\rho_i$ according to (6.12)
For $j = 0, \cdots, \rho_i - 1$, $h = 0, \cdots, \rho_i$
Step 5   Calculate $P_{i,j}(k-j)$ and $N_{i,j-1}$ in (6.89) and (6.90)
Step 6   Calculate $\Sigma_{i,h}(k-h)$ and $L_{i,h}(k-h)$ in (6.92), (6.93)
Step 7   Calculate $\bar{L}_h(z)$ in (6.110)
Step 8   Calculate $S_{i,h}$, $\bar{\Sigma}_{i,h}(k-h)$, $\bar{\Sigma}_{i,h}^{fs}(k-h)$ in (6.113), (6.115)
Step 9   Set $J_{th}$ in (6.117)
Online Detection
At time instant $k$, $j = 0, \cdots, \rho_i - 1$, $h = 0, \cdots, \rho_i$
Step 1   Calculate $\hat{\xi}_{i,j}^{o}(k-j)$ and $r_{i,h}(k-h)$ in Theorem 6.3
Step 2   Calculate $\hat{\xi}_{i,j}^{o,sf}(k-j)$ in Theorem 6.4
Step 3   Calculate $\bar{r}_{i,h}(k-h)$ and $\bar{r}_{i,h}^{fs}(k-h)$ in (6.112), (6.114)
Step 4   Calculate $J_i(k-h)$ and $J_i^{fs}(k-h)$ in (6.117)
Step 5   Make decision by (3.1)
Step 6   Save $\hat{\xi}_{i,\rho_i}^{o}(k-\rho_i+1)$ for next $k+1$
```

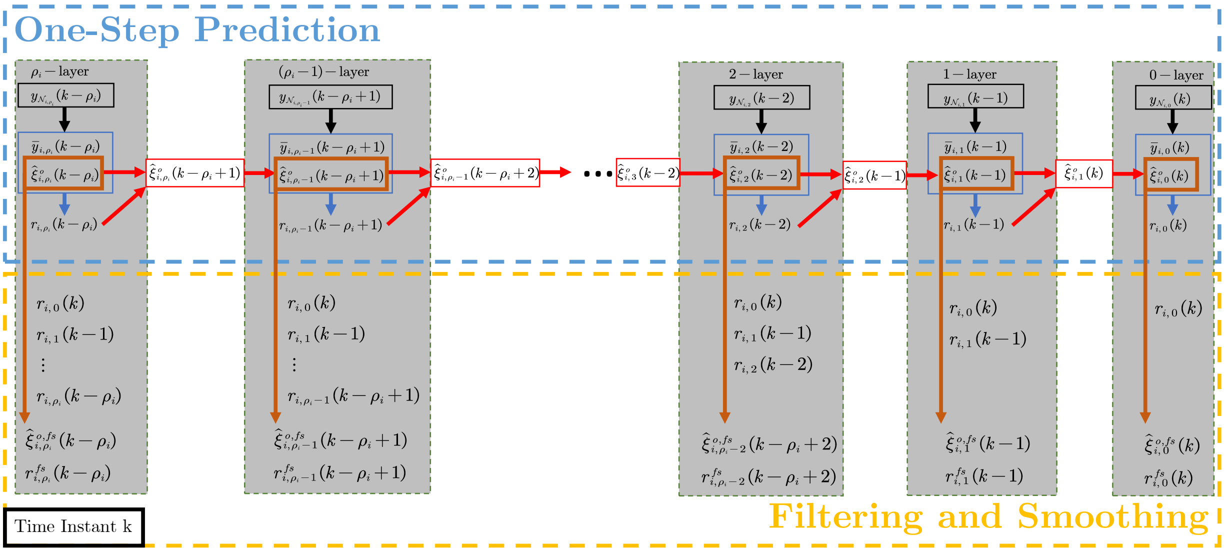

One-Step Prediction
$\rho_i -$ layer
$y_{\mathcal{N}_{i,\rho_i}}(k-\rho_i)$
$\bar{y}_{i,\rho_i}(k-\rho_i)$
$\hat{\xi}^o_{i,\rho_i}(k-\rho_i)$
$r_{i,\rho_i}(k-\rho_i)$
$\hat{\xi}^o_{i,\rho_i}(k-\rho_i+1)$
$(\rho_i-1)-$ layer
$y_{\mathcal{N}_{i,\rho_i-1}}(k-\rho_i+1)$
$\bar{y}_{i,\rho_i-1}(k-\rho_i+1)$
$\hat{\xi}^o_{i,\rho_i-1}(k-\rho_i+1)$
$r_{i,\rho_i-1}(k-\rho_i+1)$
$\hat{\xi}^o_{i,\rho_i-1}(k-\rho_i+2)$
$\hat{\xi}^o_{i,3}(k-2)$
$2-$ layer
$y_{\mathcal{N}_{i,2}}(k-2)$
$\bar{y}_{i,2}(k-2)$
$\hat{\xi}^o_{i,2}(k-2)$
$r_{i,2}(k-2)$
$\hat{\xi}^o_{i,2}(k-1)$
$1-$ layer
$y_{\mathcal{N}_{i,1}}(k-1)$
$\bar{y}_{i,1}(k-1)$
$\hat{\xi}^o_{i,1}(k-1)$
$r_{i,1}(k-1)$
$\hat{\xi}^o_{i,1}(k)$
$0-$ layer
$y_{\mathcal{N}_{i,0}}(k)$
$\bar{y}_{i,0}(k)$
$\hat{\xi}^o_{i,0}(k)$
$r_{i,0}(k)$
$r_{i,0}(k)$
$r_{i,1}(k-1)$
$\vdots$
$r_{i,\rho_i}(k-\rho_i)$
$\hat{\xi}^{o,fs}_{i,\rho_i}(k-\rho_i)$
$r^{fs}_{i,\rho_i}(k-\rho_i)$
$r_{i,0}(k)$
$r_{i,1}(k-1)$
$\vdots$
$r_{i,\rho_i-1}(k-\rho_i+1)$
$\hat{\xi}^{o,fs}_{i,\rho_i-1}(k-\rho_i+1)$
$r^{fs}_{i,\rho_i-1}(k-\rho_i+1)$
$r_{i,0}(k)$
$r_{i,1}(k-1)$
$r_{i,2}(k-2)$
$\hat{\xi}^{o,fs}_{i,\rho_i-2}(k-\rho_i+2)$
$r^{fs}_{i,\rho_i-2}(k-\rho_i+2)$
$r_{i,0}(k)$
$r_{i,1}(k-1)$
$\hat{\xi}^{o,fs}_{i,1}(k-1)$
$r^{fs}_{i,1}(k-1)$
$r_{i,0}(k)$
$\hat{\xi}^{o,fs}_{i,0}(k)$
$r^{fs}_{i,0}(k)$
Time Instant k
Filtering and Smoothing

Figure 6.3: Prediction, filtering and smoothing procedures

6.5 Examples

In this section we provide examples for proposed methods.

Static Case

Given a system modelled by (6.2) with 5 nodes and sampling time $T = 0.02s$, where

$$x = \begin{bmatrix} 1 & 2 & 3.2 & 2.3 & 5.4 \end{bmatrix}^T, \; A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \; A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \; A_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \; A_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B_1 = B_2 = B_3 = B_4 = B_5 = 1.$$

The variance matrix of measurement noise is given by

$$\Sigma = \begin{bmatrix} 16.7 & 17.5 & 5.25 & 7.2 & 11.2 \\ 17.5 & 25.824 & 14.51 & 9.62 & 17.36 \\ 5.25 & 14.51 & 26.805 & 7.07 & 23.45 \\ 7.2 & 9.62 & 7.07 & 6.4 & 6.4 \\ 11.2 & 17.36 & 23.45 & 6.4 & 25.9 \end{bmatrix}.$$

The task is to detect sensor fault in node 1 and the communication topology is shown in Fig. 6.4.

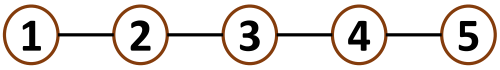


Figure 6.4: Communication topology for static case

Algebraic representations of Figure 6.4 for node 1 are $d_1 = 4$, $\mathcal{N}_{1,0} = \{1\}$, $\mathcal{N}_{1,1} = \{2\}$, $\mathcal{N}_{1,3} = \{4\}$, $\mathcal{N}_{1,4} = \{5\}$. Set $\delta = 3.5$ and solve (6.12) to obtain $\rho_1 = 3$. Set FAR $\leqslant 0.95$, we have $J_{th} = \chi^2_{0.95}(1) = 3.8415$. A step fault happening at 10s. Simulation results of the evaluation functions for $J_{1,k}(k)$, $J_{1,k+1}(k)$, $J_{1,k+2}(k)$ and $J_{1,k+3}(k)$ are shown in in Figure 6.5. It verifies the effectiveness of the proposed method. Moreover, the result shows that better detection performance is obtained via more measurement data since the different between evaluation function and threshold become larger when more measurements are involved.

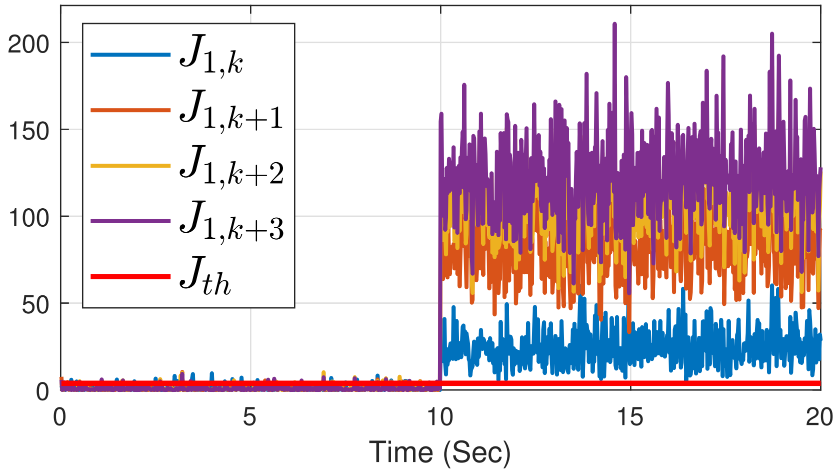


Figure 6.5: Simulation result for static case

Dynamic Case

Give system model (6.39) with

$$
\begin{aligned}
&A_{11} = 1, A_{12} = 2, C_1 = 1, R_1 = 2.4, Q_1 = 0.4,\\
&A_{21} = 1, A_{22} = 2, A_{23} = 5, C_2 = 1, R_2 = 2, Q_2 = 0.6,\\
&A_{32} = 7, A_{33} = 3, A_{34} = 2, C_3 = 1, R_3 = 1.6, Q_3 = 0.8,\\
&A_{43} = 2, A_{44} = 4, A_{45} = 3, C_4 = 1, R_4 = 1.2, Q_4 = 0.6,\\
&A_{54} = 3, A_{55} = 4, A_{56} = 5, C_5 = 1, R_5 = 0.8, Q_5 = 0.2,\\
&A_{65} = 4, A_{66} = 5, C_6 = 1, R_6 = 0.4, Q_6 = 0.8.
\end{aligned}
$$

The topology of the physical links and sensor network is shown in Figure 6.6.

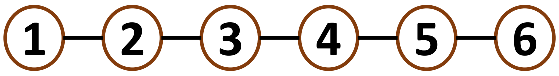


Figure 6.6: Topology of physical links and sensor network for dynamic case

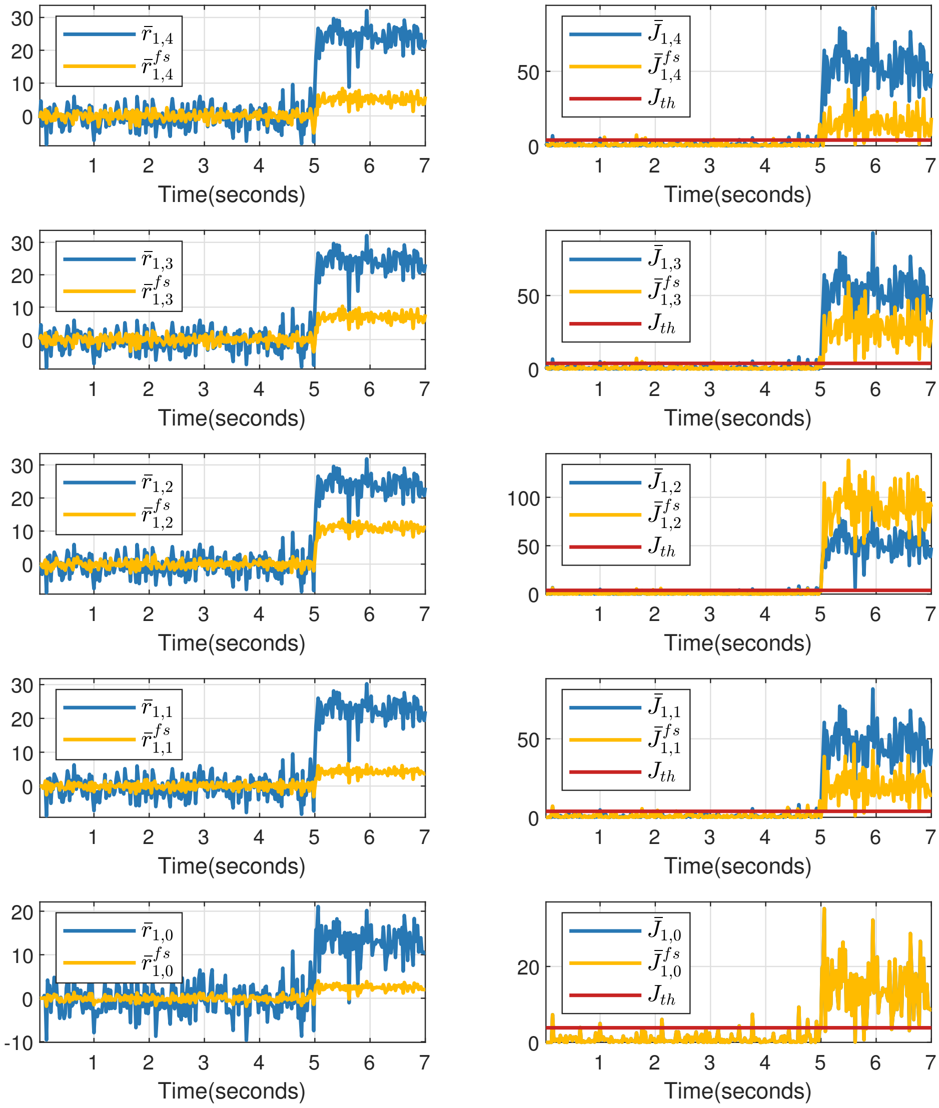


Figure 6.7: Simulation result for dynamic case

For node 1, set $\eta_{acc} = 550$, solve (6.84) to obtain $\rho_1 = 4$, so the measurements from node 1, 2, 3, 4 and 5 are applied for distributed FD in node 1. With $T = 0.02$s, FAR $= 0.95$, and an additive step fault happening at 4s in node 1, simulation results of residuals, evaluations and thresholds are shown in Figure 6.7. We can see that the amplitude of $\bar{r}^{fs}$ is smaller than $\bar{r}$ in fault-free case and the evaluations J cross the corresponding threshold J_{th} after fault happens. These indicates that our proposed distributed FD scheme can satisfy the need to detection faults in large-scale interconnected systems.

6.6 Concluding Remarks

In this chapter, a novel distributed FD scheme is proposed for large-scale interconnected systems by taking into account the influence of noise and transmission time of information exchange. With the layer structure and node selection, the approach uses local and other sensors' information to do optimal FD via optimal estimation. For static cases, it uses the correlation between different signals to reduce the uncertainty for FD. For dynamic cases, the observable subspace of local measurement is estimated by means of the one-step prediction, and then the estimation is improved by filtering and smoothing techniques. In this way, node i makes full use of data, which it receives within time k, delivers minimal variance estimation of local observable subspace, and then applies it for performing FD.

7 Benchmark Studies

This chapter provides the applications of proposed FD methods in the previous chapters to benchmark processes. We consider three benchmark processes, DC micro-grid, mass-spring system, and six-tank system. To be specific, the proposed distributed realization of optimal FD in Chapter 4 is applied for a DC micro-grid, the distributed state observer and post-filter introduced in Chapters 5 are adopted for a mass-spring system, and a six-tank system is used to show the effectiveness of the distributed FD method shown in Chapter 6.

7.1 Case Study on Micro-Grid Power Network

As a result of technological advancements, cost reductions, and proven success, micro-grids are transitioning from experimental stages to commercial markets [56]. These self-contained power systems are utilized to enhance the dependability and resilience of power grids, manage the integration of clean energy resources like wind and solar photovoltaic generation to reduce the use of fossil fuels, and provide electricity for certain purposes. A micro-grid is an independent local electrical grid that can operate either connected to the main power grid or in isolation, with defined electrical boundaries that make it a single and controllable entity. The grid can provide electricity, and potentially heating and cooling, to nearby customers using advanced software and control systems [57]. Micro-grids are able to switch between grid-connected and island modes, providing emergency power to customers even when the main grid is down. A micro-grid that is connected to the main power grid operates in grid-connected mode. However, when technical or economic conditions require, it can disconnect from the main grid and operate autonomously in island mode, improving the security of power supply within the micro-grid. An islanded micro-grid is an independent electricity system that has its own sources of power, including an energy storage system. It is also utilized in situations where power transmission and distribution from a centralized energy source are impractical due to distance and cost. For example, it offers a solution for providing electricity to remote areas and smaller islands [71]. In islanded operation mode, the energy management system monitors the power generation and consumption of the entire micro-grid in real time. This management system can also disconnect from the traditional grid and operate autonomously, making the islanded micro-grid a local energy grid with control capability. Since the reliability of the micro-grid system is important and meaningful, the implementation of FD scheme is reasonable to be considered.

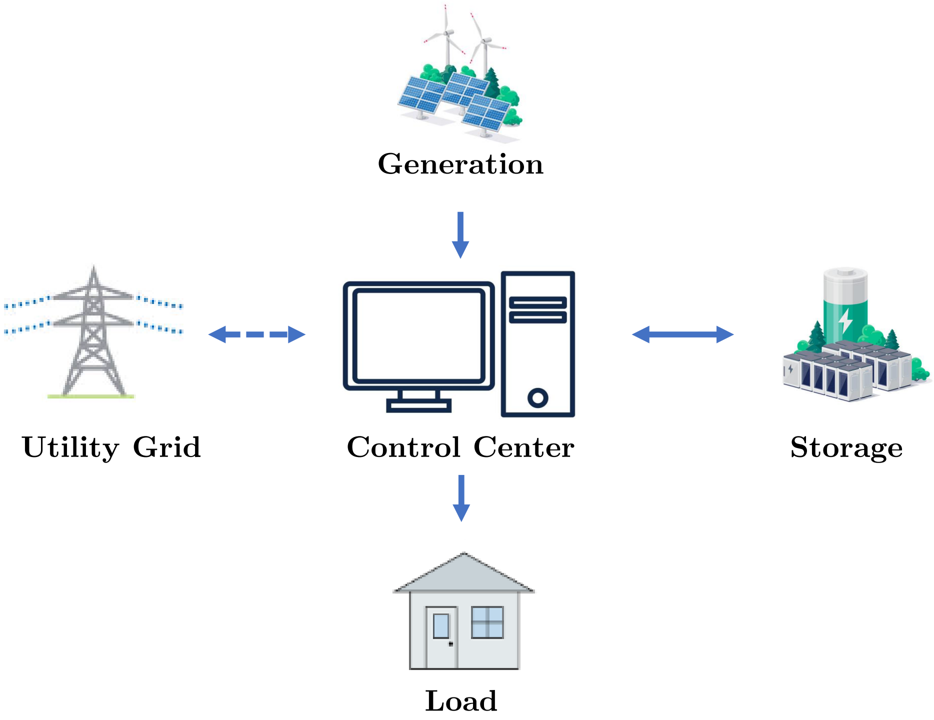


Figure 7.1: Structure of micro-grid

In this section, the model of DC micro-grid from [95] is adopted. Figure 7.2 shows

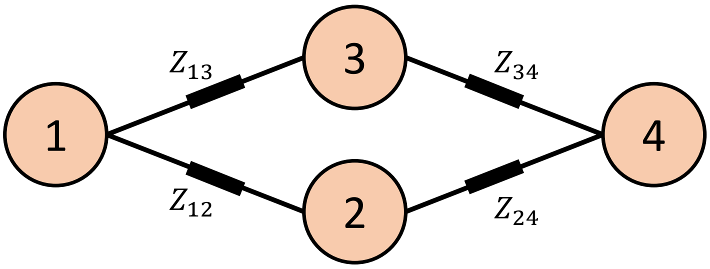


Figure 7.2: An islanded DC micro-grid consisting of 4 distributed generations

a topology for an islanded DC micro-grid composed of 4 distributed generations (DG) connected via distributed line Z_{ij}. A sensor network is adopted to monitor the process and the corresponding sate space representation is

$$\dot{x}_i = A_{ii}x_i + \sum_{j\in\mathcal{N}_i} A_{ij}x_j + B_iu_i + E_iw_i,\ \ y_i = C_ix_i + \bar{F}_id_i,\ \ i = 1,2,3,4$$

$$A_{ii} = \begin{bmatrix} -\sum_{j\in\mathcal{N}_i} \dfrac{1}{C_{t_i}R_{ij}} & \dfrac{1}{C_{t_i}} \\ -\dfrac{1}{L_{t_i}} & -\dfrac{R_{t_i}}{L_{t_i}} \end{bmatrix}, \; A_{ij} = \begin{bmatrix} \dfrac{1}{C_{t_i}R_{ij}} & 0 \\ 0 & 0 \end{bmatrix}$$

$$B_i = \begin{bmatrix} 0 \\ \dfrac{1}{L_{t_i}} \end{bmatrix}, \; E_i = \begin{bmatrix} -\dfrac{1}{C_{t_i}} \\ 0 \end{bmatrix}, \; C_i = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; \bar{F}_i = \begin{bmatrix} 1 \end{bmatrix}$$

where $x_i \in \mathcal{R}^2$ is the state of DG i, $u_i \in \mathcal{R}^1$ is input, $w_i \in \mathcal{R}^1$ is unknown external input in processes, $d_i \in \mathcal{R}^1$ is the measurement disturbance and $y_i \in \mathcal{R}^1$ is measurement of sensor i. Parameters $R_{ij} = R_{ji}$ and the values of parameters are shown in Table 7.1.

DGs	$R_t(\Omega)$	$L_t(mH)$	$C_t(mF)$	Line impedance Z_{ij}	$R_{ij}(\Omega)$
DG 1	0.2	1.8	2.2	Z_{12}	0.05
DG 2	0.3	2.0	1.9	Z_{13}	0.07
DG 3	0.1	2.2	1.7	Z_{24}	0.04
DG 4	0.5	3.0	2.5	Z_{34}	0.06

Table 7.1: Parameters of DC micro-grid

Stacking all nodes together, we have the centralized form

$$\dot{x} = A_c x + Bu + E_c d_l, \; y_l = C_l x + F_l d_l, \tag{7.1}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \; u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}, \; y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}, \; d_l = \begin{bmatrix} d_p \\ d_m \end{bmatrix}, \; d_p = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}, \; d_m = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}, \tag{7.2}$$

$$A_c = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}, B = \begin{bmatrix} B_1 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ 0 & 0 & B_3 & 0 \\ 0 & 0 & 0 & B_4 \end{bmatrix}, C_l = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{bmatrix}, \tag{7.3}$$

$$E_c = \begin{bmatrix} E_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & E_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & E_4 & 0 & 0 & 0 & 0 \end{bmatrix}, \; F_l = \begin{bmatrix} 0 & 0 & 0 & 0 & \bar{F}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{F}_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{F}_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F}_4 \end{bmatrix}. \tag{7.4}$$

Discretize the process (7.1) with a sampling interval $T = 0.01$s leading to the discrete-time model (4.6) and (4.7) with

$$A = \begin{bmatrix} 1.046 & 0.005 & -0.096 & -0.000 & -0.068 & -0.000 & 0.010 & 0.000 \\ -0.006 & 0.999 & 0.000 & 0.000 & 0.000 & 0.000 & -0.000 & -0.000 \\ -0.111 & -0.000 & 1.072 & 0.005 & 0.008 & 0.000 & -0.139 & -0.000 \\ 0.000 & 0.000 & -0.005 & 0.998 & -0.000 & -0.000 & 0.000 & 0.000 \\ -0.088 & -0.000 & 0.009 & 0.000 & 1.053 & 0.006 & -0.103 & -0.000 \\ 0.000 & 0.000 & -0.000 & -0.000 & -0.005 & 1.000 & 0.000 & 0.000 \\ 0.008 & 0.000 & -0.106 & -0.000 & -0.070 & -0.000 & 1.051 & 0.004 \\ -0.000 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.003 & 0.998 \end{bmatrix},$$

$$E_l = \begin{bmatrix} -0.005 & 0.000 & 0.000 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.000 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.005 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.000 & 0.000 & 0.000 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.000 & -0.006 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.000 & 0.000 & 0.000 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.000 & 0.000 & 0.000 & -0.004 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.000 & -0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \end{bmatrix},$$

$$F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & \bar{F}_1 & 0 & 0 & 0 \end{bmatrix}, \; F_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \bar{F}_2 & 0 & 0 \end{bmatrix},$$
$$F_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \bar{F}_3 & 0 \end{bmatrix}, \; F_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F}_4 \end{bmatrix},$$

C_l and F_l are the same as shown in (7.3) and (7.4). Notice that using (4.5) to build communication topology of sensor network leads to an unconnected graph, thus we choose the the same topology as shown in Figure 7.2. Apply Algorithm (4.2) to perform distributed FD for the DC micro-grid. Given $\delta_d = 4.3415$, we set

$$J_i(k) = \left(\frac{1}{5} \sum_{l=0}^{4} j_i(k+l) \right)^{1/2}, \; J_{th} = \sqrt{\frac{1}{5}\delta_d} = 1.9416.$$

With an actuator fault at 3s in DG 1, simulation results of residuals, evaluations and thresholds are shown in Figures 7.3 and 7.4. They indicate that our proposed distributed FD scheme can achieve a similar result when compared with the centralized approach, which is plotted as J in each figure.

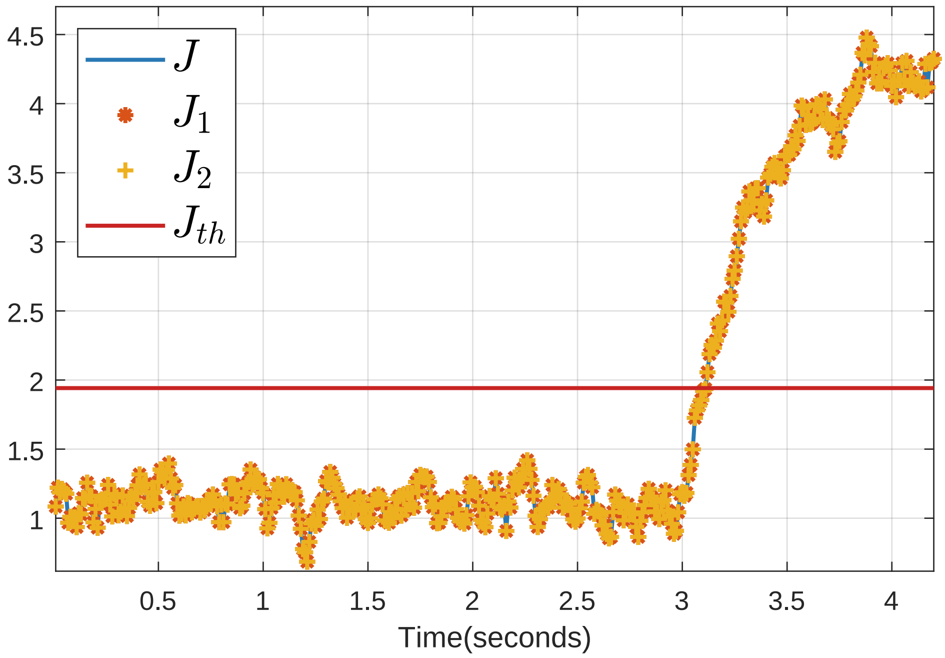


Figure 7.3: Simulation result for sensor nodes 1 and 2 in DC micro-grid

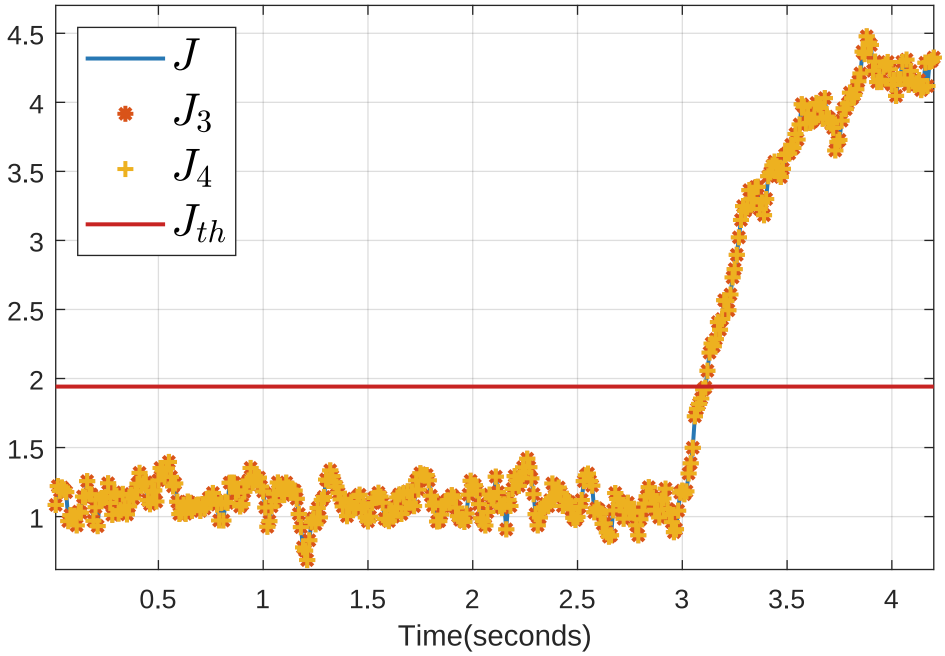


Figure 7.4: Simulation result for sensor nodes 3 and 4 in DC micro-grid

7.2 Case Study on Mass-Spring System

In this section, we introduce the benchmark of the mass-spring system illustrated in Figure 7.5. The system is composed of three cars, which are connected by springs. The physical relations between subsystems are

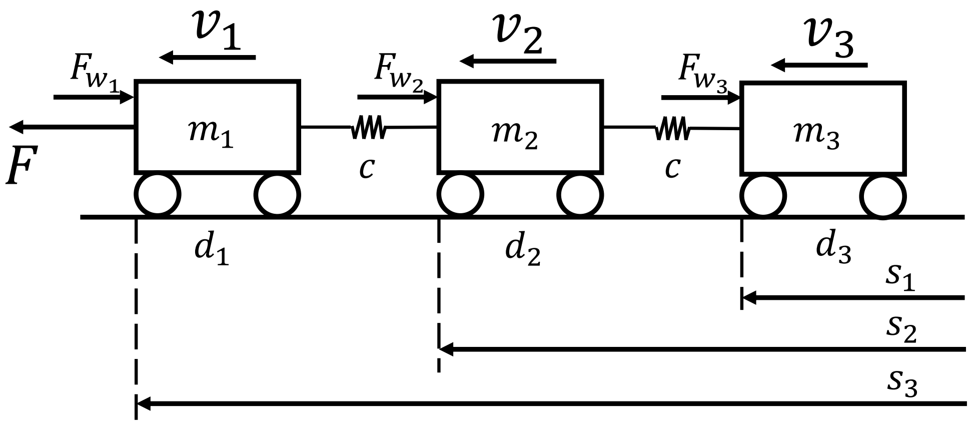


Figure 7.5: Mass-spring system

$$
\begin{aligned}
\dot{v}_1 &= \frac{F}{m_1} - \frac{d_1 v_1}{m_1} - \frac{F_{w1}}{m_1} - \frac{c}{m_1}(s_1 - s_2) \\
\dot{v}_2 &= -\frac{d_2 v_2}{m_2} - \frac{F_{w2}}{m_2} + \frac{c}{m_2}(s_1 - s_2) - \frac{c}{m_2}(s_2 - s_3) \\
\dot{v}_3 &= -\frac{d_3 v_3}{m_3} - \frac{F_{w3}}{m_3} + \frac{c}{m_3}(s_2 - s_3)
\end{aligned} \tag{7.5}
$$

where v_i is the velocity of car i, F_{wi} is unknown disturbance in car i, and s_i is the position of subsystem i. Each car is equipped with sensor to measure its velocity and position with

$$
y_i = \begin{bmatrix} v_i \\ s_i \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \bar{d}_i, i = 1, 2, 3 \tag{7.6}
$$

where $\bar{d}_i$ is the measurement disturbance. Other parameters are shown in Table 7.2. Discretize the process (7.5) and (7.6) with a sampling interval $T = 0.01$s leading to the discrete-time model (5.1) with

$$
A_{11} = \begin{bmatrix} 0.9999 & -0.0075 \\ 0.0100 & 1.0000 \end{bmatrix}, A_{12} = \begin{bmatrix} 0 & 0.0075 \\ 0 & 0 \end{bmatrix}, A_{13} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

Parameters	Symbol	Value	Unit
mass of car 1	m_1	10	kg
mass of car 2	m_2	20	kg
mass of car 3	m_3	10	kg
spring constant	c	6.5	N/m
coeff. of friction of car 1	d_1	0.1	kg/s
coeff. of friction of car 2	d_2	0.1	kg/s
coeff. of friction of car 3	d_3	0.1	kg/s

Table 7.2: Parameters of mass-spring system

$$A_{21} = \begin{bmatrix} 0 & 0.0037 \\ 0 & 0 \end{bmatrix}, A_{22} = \begin{bmatrix} 0.9999 & -0.0075 \\ 0.01 & 1 \end{bmatrix}, A_{23} = \begin{bmatrix} 0 & 0.0037 \\ 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$A_{31} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, A_{32} = \begin{bmatrix} 0 & 0.0075 \\ 0 & 0 \end{bmatrix}, A_{33} = \begin{bmatrix} 0.9999 & -0.0075 \\ 0.01 & 1 \end{bmatrix}, C_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$E_{d1} = \begin{bmatrix} -0.001 & 0 & 0 \\ -0.000005 & 0 & 0 \end{bmatrix}, F_{d1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, E_{d2} = \begin{bmatrix} -0.0005 & 0 & 0 \\ -0.0000025 & 0 & 0 \end{bmatrix}, F_{d2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$E_{d3} = \begin{bmatrix} -0.0001 & 0 & 0 \\ -0.000005 & 0 & 0 \end{bmatrix}, F_{d3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, x_i = \begin{bmatrix} v_i \\ s_i \end{bmatrix}, d_i = \begin{bmatrix} F_{wi} \\ \bar{d}_i \end{bmatrix}$$

The communication topology of the considered system is the same as its physical links as shown in Figure 7.6.

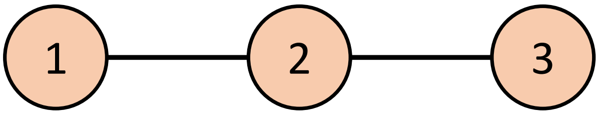


Figure 7.6: Communication topology in mass-spring system

Apply Theorem 5.1 to compute the distributed observer gains. The results are

$$L_{11} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}, L_{12} = \begin{bmatrix} 0 & 0.75 \\ 0 & 0 \end{bmatrix}, L_{13} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$L_{21} = \begin{bmatrix} 0 & 0.0037 \\ 0 & 0 \end{bmatrix}, L_{22} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}, L_{23} = \begin{bmatrix} 0 & 0.37 \\ 0 & 0 \end{bmatrix},$$

$$L_{31} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, L_{32} = \begin{bmatrix} 0 & 0.75 \\ 0 & 0 \end{bmatrix}, L_{33} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}.$$

The target for subsystem 1 is to detect local actuator fault in itself with

$$E_{f1} = \begin{bmatrix} 0.0001 \\ 0.000005 \end{bmatrix}, F_{f1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

For the post-filter design, we only consider the actuator fault at node 1 and all disturbances. Adopt Theorems 5.3 and 5.4 to design post-filter. Set $s = 2$ for model (5.18), we have

$$v_{1,2} = \begin{bmatrix} -0.2311 & -0.2129 & 0.6532 & 0.6365 & -0.4609 & -0.4747 \end{bmatrix}$$

as the result for Theorem 5.3 and

$$G_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, L_1 = -\begin{bmatrix} -0.2311 & -0.2129 \\ 0.6532 & 0.6365 \end{bmatrix}, v_1 = \begin{bmatrix} -0.4609 & -0.4747 \end{bmatrix}, w_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

as the result for Theorem 5.4. Given $\delta_d = 1.5$, further compute $\gamma_1 = 2.2204$ in (5.26) and set

$$J_1(k) = \sqrt{\frac{1}{6}\sum_{j=k-5}^{k} \epsilon_1^T(j)\epsilon_1(j)},\ J_{th,1} = \gamma_1\sqrt{\frac{1}{6}}\delta_d = 1.3597$$

in (5.27). With a step fault happens at $30s$, simulation result is shown in Figure 7.7. It can be seen that our proposed method can distributively detect the occurrence of fault.

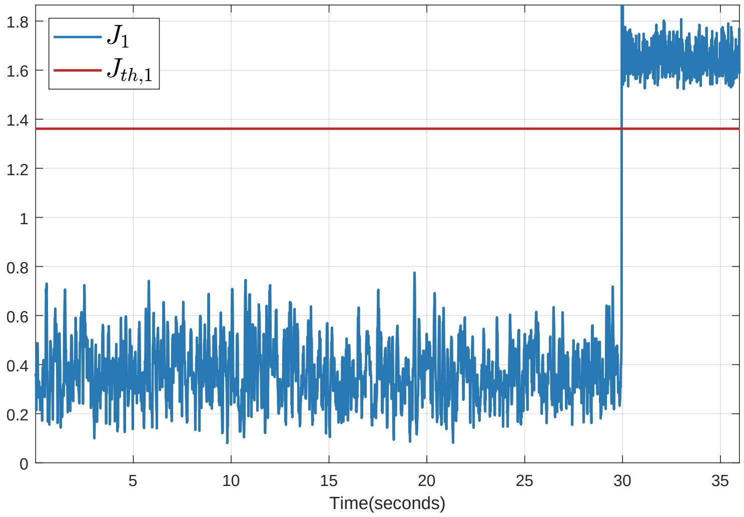


Figure 7.7: Simulation result of subsystem 1 in mass-spring system

7.3 Case Study on Six-Tank System

Tank systems, commonly utilized in the chemical industry, serve as containers for various chemicals and are available in various sizes and shapes. These tanks play a crucial role in both storing and transporting raw materials and finished chemical products, as well as in the processing and mixing of chemicals. It is important to note that proper chemical storage is vital to ensure the safety of individuals and the environment. Thus, it is important to implement effective monitoring and detection systems that can alert the occurrence of faults to further compensations.

Figure 7.8: An example of tank systems [1]

In this section, we provide a case study on six-tank system to illustrate the effectiveness of the proposed distributed FD technique for dynamic systems in Chapter 6. Consider the six-tank system as sketched in Figure 7.9.

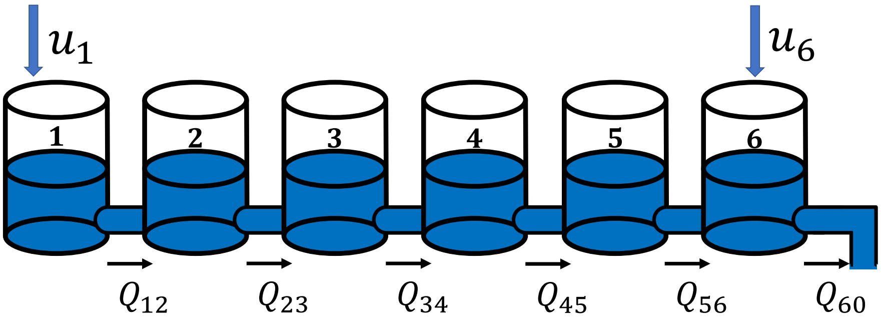


Figure 7.9: Six-tank system

The dynamics of six-tank system is modelled by

$$\mathcal{A}\dot{h}_1 = u_1 - Q_{12},\ \mathcal{A}\dot{h}_2 = Q_{12} - Q_{23},\ \mathcal{A}\dot{h}_3 = Q_{23} - Q_{34},$$
$$\mathcal{A}\dot{h}_4 = Q_{34} - Q_{45},\ \mathcal{A}\dot{h}_5 = Q_{45} - Q_{56},\ \mathcal{A}\dot{h}_6 = u_6 + Q_{56} - Q_{60}$$
$$Q_{ij} = \mathrm{sgn}(h_i - h_j)a_i s_0\sqrt{2g|h_i - h_j|},\ Q_{60} = a_6 s_0\sqrt{2gh_6}$$

[1] The figure is from https://gpi-tanksxl.com/nen-en13445/

where h_i is the water level (cm) of tank i, u_i is the 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. Parameters of six-tank system are shown in Table 7.3. After linearization at given operating point $h_1 = 40, h_2 = 30.77, h_3 = 25.58, h_4 = 20.38, h_5 = 15.19, h_6 = 10$ and discretization with sampling time $T = 0.01s$, we have the model (6.39) with 6 subsystems, the system matrices are

Parameters	Symbol	Value	Unit
cross section area of tanks	$\mathcal{A}$	154	cm^2
cross section area of pipes	s_0	0.5	cm^2
coeff. of flow for pipe 1	a_1	0.45	
coeff. of flow for pipe 2	a_2	0.60	
coeff. of flow for pipe 3	a_3	0.60	
coeff. of flow for pipe 4	a_4	0.60	
coeff. of flow for pipe 5	a_5	0.60	
coeff. of flow for pipe 6	a_6	0.45	

Table 7.3: Parameters of six-tank system

$$\begin{aligned}
&A_{11} = 0.9999, A_{12} = 0.0001, C_1 = 1,\\
&A_{21} = 0.0001, A_{22} = 0.9997, A_{23} = 0.0002, C_2 = 1,\\
&A_{32} = 0.0003, A_{33} = 0.9996, A_{34} = 0.0002, C_3 = 1,\\
&A_{43} = 0.0002, A_{44} = 0.9996, A_{45} = 0.0002, C_4 = 1,\\
&A_{54} = 0.0002, A_{55} = 0.9996, A_{56} = 0.0002, C_5 = 1,\\
&A_{65} = 0.0002, A_{66} = 0.9997, C_6 = 1.
\end{aligned}$$

And the parameters for process and measurement noises are

$$\begin{aligned}
&R_1 = 1.2, Q_1 = 1, R_2 = 1.2, Q_2 = 0.3, R_3 = 1.2, Q_3 = 0.4\\
&R_4 = 1.2, Q_4 = 0.3, R_5 = 1.2, Q_5 = 0.1, R_6 = 1.2, Q_6 = 0.4
\end{aligned}$$

The topology of the physical links and sensor network is shown in Figure 7.10.

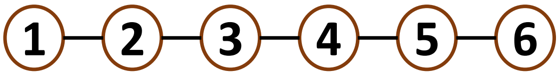


Figure 7.10: Topology of physical links and sensor network for six-tank system

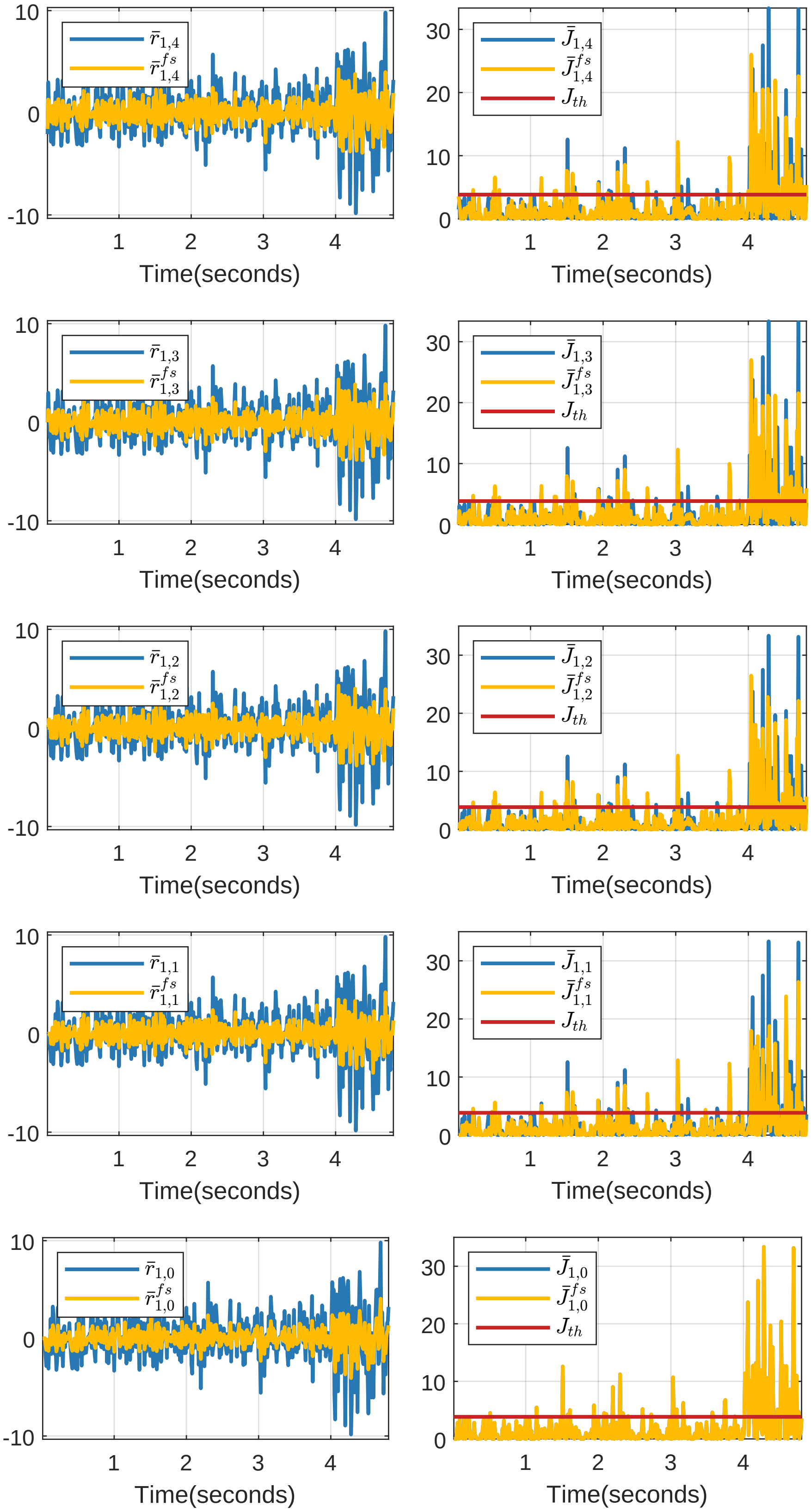


Figure 7.11: Simulation result

For node 1, set $\eta_{acc} = 550$, solve (6.12) to obtain $\rho_1 = 4$, so the measurements from node 1, 2, 3, 4 and 5 are applied for distributed FD in node 1. With FAR $= 0.95$, and a leakage happening at 4s in tank 1, simulation results of residuals, evaluations and thresholds are shown in Figure 7.11. We can see that the evaluations J cross the corresponding threshold J_{th} after fault happens. These indicates the effectiveness of the proposed distributed FD scheme.

7.4 Concluding Remarks

In this chapter, the proposed distributed FD approaches are applied in benchmark processes to demonstrate their effectiveness. The distributed realization of the centralized optimal FD method is used to detect faults in micro-grid systems. The results have shown that the proposed method achieves a similar result as the centralized optimal one. In addition, the mass-spring benchmark has been used to show the effectiveness of the combination of distributed state observer and post-filter. The six-tank process, which is a nonlinear system but linearized at a specific operating point, has been applied to demonstrate the effectiveness of distributed FD method proposed in Chapter 6. The case study results presented in this chapter satisfy the need for distributed detection of potential faults.

8 Conclusions and Future Works

The main focus of this thesis is to investigate distributed FD schemes for large-scale and interconnected systems, which have physical links between subsystems, with sensor networks. In this chapter, we summarise the main insights of this thesis and discuss future scopes.

8.1 Conclusions

In Chapter 1, we briefly introduce the primary and basic concepts of large-scale and interconnected systems, FD, and process monitoring. It has been emphasized that advanced and efficient distributed FD methods shall be investigated for large-scale and interconnected systems with the development of sensing hardware and communication techniques.

Definitions, fundamental knowledge, and mathematical preliminaries of LS estimation, graph theory, average consensus, and LMI techniques are presented in Chapter 2. The algorithm of LS estimation plays an essential role in optimal observer design, which is the core of FD, for industrial processes. The graph theory describes the model of physical links of interconnected systems and the communication topology of sensor networks. Both average consensus and LMI techniques are the designing techniques for distributed algorithms. Parallel to Chapter 2, Chapter 3 is devoted to the state of the art of FD methodologies for static and dynamic systems influenced by stochastic noises and deterministic disturbances. Besides the concepts of optimal FD problems and models of static and dynamic systems, different types of residual generators are introduced. These fundamental methodologies serve as the basis for subsequent studies.

Although the optimal FD methods have shown their effectiveness in wide industrial applications, they are mainly centralized. The objective of Chapter 4 is to develop an optimal distributed FD approach for large-scale systems in the presence of unknown deterministic disturbances using the measurement of sensor networks. With the help of the average consensus algorithm, the optimal FD method based on $\mathcal{H}_2$ observer is realized in a distributed manner, and this distributed realization shows a similar result as the original centralized one. In Chapter 5, a widely used form of the distributed observer is applied for FD for large-scale and interconnected systems. In this framework, each local observer uses only its local and neighbours' information to estimate its local state. The LMI technique is applied to design the parameters of the distributed observer. It delivers sufficient numerical solutions. Thereafter, a post-filter is applied to enhance the influence of fault on the residual signal and reduce the influence of disturbance on the residual signal. For the design of the post-filter, PSA and DO are applied.

Chapter 6 describes distributed approaches for FD in interconnected systems under the influence of random noises. Firstly, a proposed criterion determines which information is selected for online FD. Then, the selected information is employed to reduce corresponding uncertainty and carry out effective FD. For static cases, the proposed distributed FD is based on the correlation between different measurements. For dynamic cases, the core part of the distributed method is that it uses prediction, filtering, and smoothing procedures to improve the accuracy of FD by means of reducing the variance matrix of the estimation error. Both methods make full use of data, which node i received within the present time instant, to do minimal variance estimation and perform accurate FD. In Chapter 7, benchmark studies are demonstrated to support the effectiveness of proposed methods in achieving their operational and strategic goals.

8.2 Future Works

This thesis attempts to build advanced and efficient distributed FD methods for large-scale and interconnected systems. The distributed FD methods are investigated based on the accurate linear model of large-scale systems. In the future, the following topics are considered as the generalization and extension of this study:

- The proposed methods are based on system models. However, modelling large-scale systems may pose significant challenges. Based on this observation, a data-driven realization of the proposed methods requires more research attentions.

- The results achieved in this thesis are based on the linear system descriptions. And the proposed approaches are efficient if the real process is working around an operating point. Therefore, extensions of the proposed methods to the nonlinear processes are of practical importance. When the analytical solution for nonlinear systems is hard to solve, numerical methods, such as partial filters and deep learning methods, are considered.

- In interconnected systems, faults can be propagated from one subsystem to others through the physical links and thus cause severe failures. In order to prevent this after the faults have been detected, fault identification methods shall be applied to locate the faults. Then FTC systems are considered to increase plant availability, reduce the risk of safety hazards, and prevent the faults from developing into catastrophes.

Bibliography

[1] N. Afgan, M. Carvalho, P. Pilavachi, A. Tourlidakis, G. Olkhonski, and N. Martins, "An expert system concept for diagnosis and monitoring of gas turbine combustion chambers," *Applied thermal engineering*, vol. 26, no. 7, pp. 766–771, 2006.

[2] M. Ahmad and R. Mohd-Mokhtar, "A survey on model-based fault detection techniques for linear time-invariant systems with numerical analysis." *Pertanika Journal of Science & Technology*, vol. 30, no. 1, 2022.

[3] T. G. Amaral, V. F. Pires, and A. J. Pires, "Fault detection in pv tracking systems using an image processing algorithm based on pca," *Energies*, vol. 14, no. 21, p. 7278, 2021.

[4] B. D. Anderson and J. B. Moore, *Optimal filtering.* Courier Corporation, 2012.

[5] S. Banerjee, J. Deng, C. Gorse, V. Vajpayee, V. Becerra, and S. Shimjith, "Ann based sensor and actuator fault detection in nuclear reactors," in *2020 8th International Conference on Control, Mechatronics and Automation (ICCMA).* IEEE, 2020, pp. 88–94.

[6] R. V. Beard, "Failure accomodation in linear systems through self-reorganization." Ph.D. dissertation, Massachusetts Institute of Technology, 1971.

[7] R. R. Bitmead, M. R. Gevers, I. R. Petersen, and R. J. Kaye, "Monotonicity and stabilizability-properties of solutions of the riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples," *Systems & Control Letters*, vol. 5, no. 5, pp. 309–315, 1985.

[8] M. Bo, J. Zhi-nong, and W. Zhong-qing, "Development of the task-based expert system for machine fault diagnosis," in *Journal of Physics: Conference Series*, vol. 364, no. 1. IOP Publishing, 2012, p. 012043.

[9] F. Boem, R. Carli, M. Farina, G. Ferrari-Trecate, and T. Parisini, "Distributed fault detection for interconnected large-scale systems: A scalable plug & play approach," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 800–811, 2018.

[10] F. Boem, R. M. Ferrari, and T. Parisini, "Distributed fault detection and isolation of continuous-time non-linear systems," *European Journal of Control*, vol. 17, no. 5-6, pp. 603–620, 2011.

[11] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press, 2022.

[12] R. J. Caverly and J. R. Forbes, "Lmi properties and applications in systems, stability, and control theory," *arXiv preprint arXiv:1903.08599*, 2019.

[13] F. Chen and W. Ren, *Distributed average tracking in multi-agent systems.* Springer, 2020.

[14] H. Chen, Z. Chen, Z. Chai, B. Jiang, and B. Huang, "A single-side neural network-aided canonical correlation analysis with applications to fault diagnosis," *IEEE Transactions on Cybernetics*, 2021.

[15] H. Chen, B. Jiang, N. Lu, and Z. Mao, "Deep pca based real-time incipient fault detection and diagnosis methodology for electrical drive in high-speed trains," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4819–4830, 2018.

[16] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems.* Springer Science & Business Media, 2012, vol. 3.

[17] T. Chen, D. J. Hill, and C. Wang, "Distributed fast fault diagnosis for multimachine power systems via deterministic learning," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 4152–4162, 2019.

[18] Z. Chen, "Data-driven fault detection for industrial processes," in *Journal of Process Control.* Springer, 2017, vol. 1.

[19] Z. Chen, Y. Cao, S. X. Ding, K. Zhang, T. Koenings, T. Peng, C. Yang, and W. Gui, "A distributed canonical correlation analysis-based fault detection method for plant-wide process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2710–2720, 2019.

[20] Z. Chen, S. X. Ding, K. Zhang, C. Yang, and T. Peng, "Generalized cca with applications for fault detection and estimation," in *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS).* IEEE, 2018, pp. 545–550.

[21] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on automatic control*, vol. 29, no. 7, pp. 603–614, 1984.

[22] W. H. Chung, J. L. Speyer, and R. H. Chen, "A decentralized fault detection filter," *J. Dyn. Sys., Meas., Control*, vol. 123, no. 2, pp. 237–247, 2001.

[23] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal of parallel and distributed computing*, vol. 7, no. 2, pp. 279–301, 1989.

[24] S. X. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools.* Springer Science & Business Media, 2008.

[25] ——, *Data-driven design of fault diagnosis and fault-tolerant control systems.* Springer, 2014.

[26] ——, "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. 24, no. 2, pp. 431–449, 2014.

[27] ——, *Advanced methods for fault diagnosis and fault-tolerant control.* Springer, 2021.

[28] S. X. Ding, Y. Yang, Y. Zhang, and L. Li, "Data-driven realizations of kernel and image representations and their application to fault detection and control system design," *Automatica*, vol. 50, no. 10, pp. 2615–2623, 2014.

[29] S. X. Ding, P. Zhang, P. M. Frank, and E. L. Ding, "Threshold calculation using lmi-technique and its integration in the design of fault detection systems," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 1. IEEE, 2003, pp. 469–474.

[30] S. X. Ding, P. Zhang, A. Naik, E. L. Ding, and B. Huang, "Subspace method aided data-driven design of fault detection and isolation systems," *Journal of process control*, vol. 19, no. 9, pp. 1496–1510, 2009.

[31] S. X. Ding, P. Zhang, S. Yin, and E. L. Ding, "An integrated design framework of fault-tolerant wireless networked control systems for industrial automatic control applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 462–471, 2012.

[32] S. Ding, "Optimal fault detection and estimation: A unified scheme and least squares solutions," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 465–472, 2018.

[33] S. Ding, E. Ding, and T. Jeinsch, "An approach to analysis and design of observer and parity relation based fdi systems," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 7718–7723, 1999.

[34] X. Ding and L. Guo, “An approach to time domain optimization of observer-based fault detection systems,” *International Journal of Control*, vol. 69, no. 3, pp. 419–442, 1998.

[35] X. Ding, L. Guo, and T. Jeinsch, “A characterization of parity space and its application to robust fault detection,” *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 337–343, 1999.

[36] E. Dubrova, *Fault-tolerant design.* Springer, 2013.

[37] R. Dunia and S. J. Qin, “Joint diagnosis of process and sensor faults using principal component analysis,” *Control Engineering Practice*, vol. 6, no. 4, pp. 457–469, 1998.

[38] B. d. et dAnalyses *et al.*, “Final report on the accident on 1st june 2009 to the airbus a330-203 registered f-gzcp operated by air france flight af 447 rio de janeiro–paris,” *Paris: BEA*, 2012.

[39] R. M. Ferrari, T. Parisini, and M. M. Polycarpou, “Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach,” *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 275–290, 2011.

[40] R. Fezai, K. Abodayeh, M. Mansouri, A. Kouadri, M.-F. Harkat, H. Nounou, M. Nounou, and H. Messaoud, “Reliable fault detection and diagnosis of large-scale nonlinear uncertain systems using interval reduced kernel pls,” *IEEE Access*, vol. 8, pp. 78 343–78 353, 2020.

[41] P. M. Frank, “Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results,” *automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[42] P. Gahinet, A. Nemirovski, and A. J. Laub, “Mahmoud chilali lmi control toolbox users guide, version 1, the mathworks,” *Inc, Natick MA,(May 1995).*

[43] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniquespart i: Fault diagnosis with model-based and signal-based approaches,” *IEEE transactions on industrial electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[44] ——, “A survey of fault diagnosis and fault-tolerant techniquespart ii: Fault diagnosis with knowledge-based and hybrid/active approaches,” 2015.

[45] J. Gertler and D. Singer, “A new structural framework for parity equation-based failure detection and isolation,” *Automatica*, vol. 26, no. 2, pp. 381–388, 1990.

[46] J. J. Gertler and R. Monajemy, “Generating directional residuals with dynamic parity relations,” *Automatica*, vol. 31, no. 4, pp. 627–635, 1995.

[47] J. L. Godoy, J. R. Vega, and J. L. Marchetti, “A fault detection and diagnosis technique for multivariate processes using a pls-decomposition of the measurement space,” *Chemometrics and Intelligent Laboratory Systems*, vol. 128, pp. 25–36, 2013.

[48] C. Godsil and G. F. Royle, *Algebraic graph theory.* Springer Science & Business Media, 2001, vol. 207.

[49] G. H. Golub and C. F. van Loan, *Matrix Computations*, 4th ed. JHU Press, 2013.

[50] L. Guan, Y. Shao, F. Gu, B. Fazenda, and A. Ball, “Gearbox fault diagnosis under different operating conditions based on time synchronous average and ensemble empirical mode decomposition,” in *2009 ICCAS-SICE.* IEEE, 2009, pp. 383–388.

[51] S. Gupta, T. Patel, C. Engelmann, and D. Tiwari, “Failures in large scale systems: long-term measurement, analysis, and implications,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017, pp. 1–12.

[52] F. Gustafsson and F. Gustafsson, *Adaptive filtering and change detection.* Wiley New York, 2000, vol. 1.

[53] W. Hackbusch, *Iterative solution of large sparse systems of equations.* Springer, 1994, vol. 95.

[54] W. K. Härdle and L. Simar, *Applied multivariate statistical analysis.* Springer Nature, 2019.

[55] F. Harrou, M. N. Nounou, H. N. Nounou, and M. Madakyaru, “Pls-based ewma fault detection strategy for process monitoring,” *Journal of Loss Prevention in the Process Industries*, vol. 36, pp. 108–119, 2015.

[56] A. Hirsch, Y. Parag, and J. Guerrero, “Microgrids: A review of technologies, key drivers, and outstanding issues,” *Renewable and sustainable Energy reviews*, vol. 90, pp. 402–411, 2018.

[57] J. Hu and A. Lanzon, “Distributed finite-time consensus control for heterogeneous battery energy storage systems in droop-controlled microgrids,” *IEEE Transactions on smart grid*, vol. 10, no. 5, pp. 4751–4761, 2018.

[58] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, “A survey of fault detection, isolation, and reconfiguration methods,” *IEEE transactions on control systems technology*, vol. 18, no. 3, pp. 636–653, 2009.

[59] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance.* Springer Science & Business Media, 2005.

[60] S. U. Jan, Y. D. Lee, and I. S. Koo, “A distributed sensor-fault detection and diagnosis framework using machine learning,” *Information Sciences*, vol. 547, pp. 777–796, 2021.

[61] Q. Jiang, S. X. Ding, Y. Wang, and X. Yan, “Data-driven distributed local fault detection for large-scale processes based on the ga-regularized canonical correlation analysis,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8148–8157, 2017.

[62] Q. Jiang, X. Yan, and B. Huang, “Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 377–386, 2015.

[63] H. L. Jones, “Failure detection in linear systems.” Ph.D. dissertation, Massachusetts Institute of Technology, 1973.

[64] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation.* Prentice Hall, 2000, no. BOOK.

[65] H. M. Khalid, A. Khoukhi, and F. M. Al-Sunni, “Fault detection and classification using kalman filter and genetic neuro-fuzzy systems,” in *2011 Annual Meeting of the North American Fuzzy Information Processing Society.* IEEE, 2011, pp. 1–6.

[66] M. Kinnaert and J. Hao, “Distributed sensor fault detection and isolation over network,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 458–11 463, 2014.

[67] D. V. Kodavade and S. D. Apte, “A universal object oriented expert system frame work for fault diagnosis,” 2012.

[68] N. Komaroff, “Iterative matrix bounds and computational solutions to the discrete algebraic riccati equation,” *IEEE Transactions on Automatic Control*, vol. 39, no. 8, pp. 1676–1678, 1994.

[69] Y. Lei, J. Lin, Z. He, and M. J. Zuo, "A review on empirical mode decomposition in fault diagnosis of rotating machinery," *Mechanical systems and signal processing*, vol. 35, no. 1-2, pp. 108–126, 2013.

[70] Y. Lei, J. Lin, M. J. Zuo, and Z. He, "Condition monitoring and fault diagnosis of planetary gearboxes: A review," *Measurement*, vol. 48, pp. 292–305, 2014.

[71] F. Li, R. Li, and F. Zhou, *Microgrid technology and engineering application.* Elsevier, 2015.

[72] L. Li, M. Chadli, S. X. Ding, J. Qiu, and Y. Yang, "Diagnostic observer design for t–s fuzzy systems: Application to real-time-weighted fault-detection approach," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 805–816, 2017.

[73] L. Li, S. X. Ding, and X. Peng, "Distributed data-driven optimal fault detection for large-scale systems," *Journal of Process Control*, vol. 96, pp. 94–103, 2020.

[74] L. Li and K. Peng, "Diagnostic observer-based fault detection approach for ts fuzzy systems," in *2019 9th International Conference on Information Science and Technology (ICIST).* IEEE, 2019, pp. 268–272.

[75] W. Li, W. Gui, Y. Xie, and S. X. Ding, "Decentralized fault detection system design for large-scale interconnected system," *IFAC Proceedings Volumes*, vol. 42, no. 8, pp. 816–821, 2009.

[76] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508).* IEEE, 2004, pp. 284–289.

[77] Y. Ma, A. Maqsood, D. Oslebo, and K. Corzine, "Comparative analysis of data driven fault detection using wavelet and fourier transform for dc pulsed power load in the all-electric ship," in *2021 IEEE Applied Power Electronics Conference and Exposition (APEC).* IEEE, 2021, pp. 1343–1347.

[78] J. F. MacGregor, C. Jaeckle, C. Kiparissides, and M. Koutoudi, "Process monitoring and diagnosis by multiblock pls methods," *AIChE Journal*, vol. 40, no. 5, pp. 826–838, 1994.

[79] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control engineering practice*, vol. 3, no. 3, pp. 403–414, 1995.

[80] K. Martin, “A review by discussion of condition monitoring and fault diagnosis in machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 34, no. 4, pp. 527–551, 1994.

[81] P. McFadden and M. Toozhy, “Application of synchronous averaging to vibration monitoring of rolling element bearings,” *Mechanical Systems and Signal Processing*, vol. 14, no. 6, pp. 891–906, 2000.

[82] J. McNames, “Fourier series analysis of epicyclic gearbox vibration,” *J. Vib. Acoust.*, vol. 124, no. 1, pp. 150–153, 2002.

[83] J. Meiry, “Failure accomodation in linear systems through self-reorganization,” *Ph. D. dissertation, Dep. Aero. Astro., Mass. Inst. Technol.*, 1971.

[84] T. K. Moon and W. C. Stirling, *Mathematical methods and algorithms for signal processing*, 2000, no. 621.39: 51 MON.

[85] A. S. Naik, “Subspace based data-driven designs of fault detection systems,” Ph.D. dissertation, Duisburg, Essen, Univ., Diss., 2010, 2010.

[86] R. Olfati-Saber, “Distributed kalman filtering for sensor networks,” in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5492–5498.

[87] J. O’Reilly, *Observers for linear systems.* Academic press, 1983, vol. 170.

[88] A. Pandey and N. H. Younan, “Underground cable fault detection and identification via fourier analysis,” in *2010 International Conference on High Voltage Engineering and Application*. IEEE, 2010, pp. 618–621.

[89] R. J. Patton and J. Chen, “A review of parity space approaches to fault diagnosis,” *IFAC Proceedings Volumes*, vol. 24, no. 6, pp. 65–81, 1991.

[90] G. Pipeleers, B. Demeulenaere, J. Swevers, and L. Vandenberghe, “Extended lmi characterizations for stability and performance of linear systems,” *Systems & Control Letters*, vol. 58, no. 7, pp. 510–518, 2009.

[91] J. Ren, “Lmi-based fault detection filter design for a class of neutral system with time delay in states,” in *2006 6th World Congress on Intelligent Control and Automation*, vol. 2. IEEE, 2006, pp. 5581–5585.

[92] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control.* Springer, 2008, vol. 27, no. 2.

[93] W. Ren and Y. Cao, *Distributed coordination of multi-agent networks: emergent problems, models, and issues.* Springer, 2011, vol. 1.

[94] S. Riverso, M. Farina, R. Scattolini, and G. Ferrari-Trecate, "Plug-and-play distributed state estimation for linear systems," in *52nd IEEE Conference on Decision and Control.* IEEE, 2013, pp. 4889–4894.

[95] M. S. Sadabadi, Q. Shafiee, and A. Karimi, "Plug-and-play robust voltage control of dc microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6886–6896, 2017.

[96] Z. B. Sahri, U. T. Malaysia *et al.*, "Support vector machine-based fault diagnosis of power transformer using k nearest-neighbor imputed dga dataset," *Journal of Computer and Communications*, vol. 2, no. 09, p. 22, 2014.

[97] A. H. A. Sari, *Data-driven design of fault diagnosis systems: Nonlinear multimode processes.* Springer Science & Business, 2014.

[98] C. Scherer and S. Weiland, "Linear matrix inequalities in control," *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands*, vol. 3, no. 2, 2000.

[99] S. Schneider, N. Weinhold, S. Ding, and A. Rehm, "Parity space based fdi-scheme for vehicle lateral dynamics," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.* IEEE, 2005, pp. 1409–1414.

[100] C. Scott Gunderson CSP, "Recognizing catastrophic incident warning signs in the process industries," *Professional Safety*, vol. 58, no. 7, p. 24, 2013.

[101] Y. Shatnawi and M. Al-Khassaweneh, "Fault diagnosis in internal combustion engines using extension neural network," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1434–1443, 2013.

[102] A. Toffolo and A. Lazzaretto, "Energy system diagnosis by a fuzzy expert system with genetically evolved rules," *International Journal of Thermodynamics*, vol. 11, no. 3, pp. 115–121, 2008.

[103] M. Valtierra-Rodriguez, R. de Jesus Romero-Troncoso, R. A. Osornio-Rios, and A. Garcia-Perez, "Detection and classification of single and combined power quality disturbances using neural networks," *IEEE transactions on industrial electronics*, vol. 61, no. 5, pp. 2473–2482, 2013.

[104] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies," *Computers & chemical engineering*, vol. 27, no. 3, pp. 313–326, 2003.

[105] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[106] H. Wang, J. Lam, S. X. Ding, and M. Zhong, "Iterative linear matrix inequality algorithms for fault detection with unknown inputs," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 219, no. 2, pp. 161–172, 2005.

[107] Y. Wang, D. Ling, W. Yang, B. Tao, and Y. Zheng, "A fault detection method with ensemble empirical mode decomposition and support vector data description," in *2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS).* IEEE, 2019, pp. 489–494.

[108] X. Wen and L. You, "A novel rolling bearing fault detection method based on wavelet transform and empirical mode decomposition," in *2019 Chinese Control Conference (CCC).* IEEE, 2019, pp. 5024–5027.

[109] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical systems and signal processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[110] R. J. Wilson, *Introduction to Graph Theory uPDF eBook.* Pearson Higher Ed, 2015.

[111] J. Wunnenberg, "Observer-based fault detection in dynamic systems=, beobachtergestutzte fehlerdetektion in dynamischen systemen," Ph.D. dissertation, Universitat-Gesamthochschule-Duisburg, 1990.

[112] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[113] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of parallel and distributed computing*, vol. 67, no. 1, pp. 33–46, 2007.

[114] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying metropolis weights," *Automatica*, vol. 1, 2006.

[115] S. Yin, “Data-driven design of fault diagnosis systems,” Ph.D. dissertation, Duisburg, Essen, Universität Duisburg-Essen, Diss., 2012, 2012.

[116] S. Yin, X. Gao, H. R. Karimi, and X. Zhu, “Study on support vector machine-based fault detection in tennessee eastman process,” in *Abstract and Applied Analysis*, vol. 2014. Hindawi, 2014.

[117] M. Yong, X. Zheng, Y. Zheng, S. Youxian, and W. Zheng, “Fault diagnosis based on fuzzy support vector machine with parameter tuning and feature selection,” *Chinese Journal of Chemical Engineering*, vol. 15, no. 2, pp. 233–239, 2007.

[118] S. Yoon and J. F. MacGregor, “Statistical and causal model-based approaches to fault detection and isolation,” *AIChE Journal*, vol. 46, no. 9, pp. 1813–1824, 2000.

[119] J. Zhang and L. Li, “An optimal distributed fault detection scheme for large-scale systems with deterministic disturbances,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 114–119, 2020.

[120] K. Zhang, K. Peng, S. X. Ding, Z. Chen, and X. Yang, “A correlation-based distributed fault detection method and its application to a hot tandem rolling mill process,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2380–2390, 2019.

[121] K. Zhang, B. Jiang, M. Chen, and X.-G. Yan, “Distributed fault estimation and fault-tolerant control of interconnected systems,” *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1230–1240, 2019.

[122] K. Zhang, B. Jiang, and V. Cocquempot, “Distributed fault estimation observer design for multi-agent systems with switching topologies,” *IET Control Theory & Applications*, vol. 11, no. 16, pp. 2801–2807, 2017.

[123] K. Zhang, B. Jiang, and P. Shi, “Distributed fault estimation observer design with adjustable parameters for a class of nonlinear interconnected systems,” *IEEE transactions on cybernetics*, vol. 49, no. 12, pp. 4219–4228, 2018.

[124] L. Zhang and N. Hu, “Time domain synchronous moving average and its application to gear fault detection,” *IEEE Access*, vol. 7, pp. 93 035–93 048, 2019.

[125] P. Zhang and S. Ding, “On fault detection in linear discrete-time, periodic, and sampled-data systems,” *Journal of control science and engineering*, vol. 2008, 2008.

[126] Q. Zhang and X. Zhang, "Distributed sensor fault diagnosis in a class of interconnected nonlinear uncertain systems," *Annual Reviews in Control*, vol. 37, no. 1, pp. 170–179, 2013.

[127] X. Zhang, *Matrix analysis and applications.* Cambridge University Press, 2017.

[128] Q. Zheng, Q. Zhao, W. Nan, and C. Li, "Oil spill in the gulf of mexico and spiral vortex," *Acta Oceanologica Sinica*, vol. 29, no. 4, p. 1, 2010.

[129] K. Zhou and J. C. Doyle, *Essentials of robust control.* Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[130] J. Zhu, Z. Ge, and Z. Song, "Distributed parallel pca for modeling and monitoring of large-scale plant-wide processes with big data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1877–1885, 2017.

DOI: 10.17185/duepublico/81272
URN: urn:nbn:de:hbz:465-20231124-162931-5