Distributed Detection of Faults and Cyber-attacks of Multi-Agent Systems

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

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

genehmigte Dissertation

von

Deyu Zhang

aus

Shandong, V.R. China

Gutachter: Prof.-Ing. Steven X. Ding
Gutachter: Prof.-Ing. Linlin. Li
Tag der mündlichen Prüfung: 30.1.2024

DOI: 10.17185/duepublico/82816
URN: urn:nbn:de:hbz:465-20250103-094428-6

Preface

This thesis was written while the author was working at the Institute for Automatic Control and Complex Systems (AKS) in the Faculty of Engineering at the University of Duisburg-Essen, Germany. First and foremost, I would like to express my sincere gratitude to my distinguished and respected supervisor, Prof. Dr.-Ing. Steven X. Ding, for his constant guidance, understanding, encouragement, and patience throughout my Ph.D. studies. He is the one who gives me inspiration for my research area, i.e. multi-agent systems. He is the one who enlightens me on empirical methods and instructs me in analyzing approaches. He is also the one who warms my heart during my academic and personal struggles. Without him, it would be impossible for the accomplishment of my thesis. I would also particularly like to give my sincere appreciation to Prof. Dr.-Ing. Linlin Li and Prof. Dr.-Ing. Changbin Hu for there productive discussions, warm encouragement and valuable suggestions.

I want to express my gratitude to my colleagues at AKS including Dr.-Ing.Birgit Kppen-Seliger, Dr.-Ing. Chris Louen, Dr.-Ing. Changsheng Hua, Dr.-Ing. Yunsong Xu, Dr.-Ing. Jiarui Zhang, Dr.-Ing. Ting Xue, Dr.-Ing. Yuhong Na, M.Sc. Tianyu Liu, M.Sc. Caroline Charlotte Zhu, M.Sc. Yannian Liu, M.Sc. Micha Obergfell, Dr.-Ing. Hogir Rafiq, M.Sc. Abdul Salah, and M.Sc. Tieqiang Wang for their excellent cooperation, valuable discussions, and helpful suggestions. Your support and contributions have been crucial to my thesis, and I am appreciative of the opportunity to have worked with you. And I owe a great debt of gratitude to Mrs. Sabine Bay, Dipl.-Ing. Klaus Gbel, Mr. Ulrich Janzen, and M.Sc. Michael Baumann for their assistance and support with our organizational duties.

Lastly, I would like to dedicate this work to my family, especially to my parents for their unconditional love, care, understanding and support.

Duisburg, 29.09.2023 Deyu Zhang

To my parents

Contents

List of Figures

List of Tables

List of Notations

Abbreviations

Abbreviation	Expansion
CPSs	cyber-physical systems
DoS	denial-of-service
FAR	false alarm rate
FDR	fault detection rate
FD	fault detection
FDF	fault detection filter
FDI	fault detection and isolation
FTC	fault-tolerant control
GLR	generalized likelihood ratio
KF	Kalman filter
LCF	left-coprime factorization
LTI	linear time-invariant
LR	likelihood ratio
MASs	multi-agent systems
MDR	missed detection rate
PFTC	passive fault tolerant control
RCF	right-coprime factorization
SKR	stable kernel representation
SIR	stable image representation
TCP	transmission control protocol
UIOs	unknown input observers
WSNs	wireless sensor networks

Mathematical notations

Notation	Description
$\mathcal{G}$	graph
$\mathcal{V}$	node set of graph
$\mathcal{E}$	edge set of graph
$\mathcal{N}_i$	The neighborhood set of node i
$\mathcal{N}_i^{(\xi)}$	The ξ-step neighborhood set of node i
$\mathcal{A}$	adjacent matrix
$\mathcal{A}_0$	adjacent matrix in attack-free case
$\mathcal{A}_f$	adjacent matrix under attacks
$\mathcal{D}$	degree matrix
$\mathcal{L}$	laplacian matrix
$\mathcal{L}_0$	laplacian matrix in attack-free case
$\mathcal{L}_f$	laplacian matrix under attacks
λ	eigenvalue of the laplacian matrix
$\forall$	for all
$\in$	belong to
$\Longrightarrow$	imply
$\Longleftrightarrow$	equivalent to
$\otimes$	kronecker product
Σ_ν	variance of ν
$\hat{\mathbf{x}}$	estimate of the state vector $\mathbf{x}$
$\mathbf{x}$	a vector
$\mathbf{X}$	a matrix
$\mathbf{X}^T$	transpose of $\mathbf{X}$
$\mathbf{X}^{-1}$	inverse of $\mathbf{X}$
$\mathbf{X} > \mathbf{0}$	$\mathbf{X}$ is positive definite matrix
$\|\|\mathbf{x}\|\|_W^2$	H_2 norm of vector $\mathbf{x}$ with a weighting W, i.e. x^TWx
I_m	identity matrix with m degree
$\mathcal{R}^n$	space of n-dimensional vectors
$\mathcal{R}^{n\times m}$	space of n by m matrices
J	evaluation function or cost function
$J_{i,l}^{\rho}$	the cost function of node i with ρ-time iteration at time instant l
J_{th}	threshold
$G(z)$	transfer matrix of a LTI system
$\mathcal{RH}_\infty$	the set of all stable transfer matrices
$\frac{\partial J(x,d)}{\partial d}$	the partial derivative of the cost function J with respect to d

$N(0, \Sigma)$	normal distribution with zero mean and Σ variance
$\chi^2(m)$	chi square distribution with m degrees of freedom
α	upper-bound of false alarm rate
δ	confidence lever

1 Introduction

This chapter briefly describes the background and motivation, objectives and outline of this thesis.

1.1 Background and Motivation

This thesis deals with distributed detection of faults and cyber-attacks in multi-agent systems (MASs). The background and motivation of this study will be introduced consecutively.

Why Investigation on Multi-agent Systems?
MASs are composed of multiple agents, which allow the agents to communicate and coordinate with their neighbors to execute complicated tasks interactively [60, 24, 5]. Compared with traditional single-agent systems, due to their inherent ability to learn and to cooperate in autonomous decision-making [81, 5, 3, 75], MASs have several advantages as follows:

- they are capable to tackle issues that a single agent cannot handle owing to a lack of ability, knowledge or resources.
- through the coordinated control and collaborative operation of the intelligent group, they vastly exceed the sum of individual performance.
- they are more scalable and upgradeable, as well as more reliable in task execution.
- they use asynchronous parallel operations among agents to increase the quality and efficiency of complicated challenges.
- they disperse their data and resources throughout systems, representing the distribution of system description problems.

With the rapid development of communication, perception, and computation technologies, MASs have received increasing attention from scholars in different disciplines, and have been widely applied to various large-scale complex engineering systems,

including but not limited to, collaborative information fusion in wireless sensor networks [51, 4, 85, 84], formation and cooperative control of multi-robot systems [37, 65, 47, 80], distributed power generation systems [55, 56, 52, 25, 78, 54], and intelligent transportation systems [48, 89, 93, 50]. This thesis will present the application of the omnidirectional mobile robotic system "Robonito" which is manufactured by the German company "Festo Didactic" [59, 1]. Several series of this mobile robot system are produced and used for education, and training, but also scientific research purposes. Robotino possesses various types of sensors, actuators, and software interfaces which are at the highest level in the field of mobile robotics. In our lab, we have two Robotino robots. Involving several virtual robots in a simulation environment called "RobotinoSim", an MAS is established by using wireless information communication.

However, MASs are typical distributed systems, which are far more complicated than centralized systems. In this regard, the requirement for system reliability is getting more critical and urgent [67]. Because of the framework of interaction with information among agents and the lack of a centralized entity, MASs are extremely vulnerable to faults and cyber-attacks [71]. Furthermore, faults or attacks that occur on any agent may not only impact that agent but also propagate via the communication network to endanger the entire system. The situation becomes considerably worse in the presence of multiple faults. In cooperative sensor networks, for example, a sensor failure or a communication error on any node in the group might cause information fusion to fail. In addition, a malfunctioning mobile agent in a formation control might threaten the global performance of MASs, causing all agents to become trapped or misled by the problematic agent. It is not surprising that detection of faults and cyber-attacks as early as possible is of great significance for MASs.

Strongly motivated by these benefits and driven by tremendous industrial demands for MASs, this thesis is dedicated to investigating the two scenarios below:

- study of a distributed fault detection algorithm via a wireless sensor network, where the sensor embedded on each moving robot is a web camera with low pixel quality.
- cyber-attack detection on cooperative multiple wheeled-robot systems, taking into account both topological changes and integrity cyber-attacks by adversaries.

Why Distributed Observer-based Fault Detection of MASs?

According to [30], a fault is defined as an unpermitted divergence of at least one system characteristic attribute or parameter from the acceptable or standard state, such as an

actuator being blocked, a sensor being lost, or a system component being disconnected. In the past two decades, it has witnessed tremendous development in the field of fault diagnosis and isolation (FDI), see [16, 19, 90] and references therein. Based on physical and mathematical knowledge of industrial systems, the model-based FDI methods have received considerable attention and found a great number of applications [30, 38]. With the significant improvement of the techniques for processing routing data, data-driven approaches are widely applied for FDI, due to their simple structure and lower needs for design and engineering effort [79, 18, 64]. Model-based FDI, on the other hand, is a more efficient and powerful tool for investigating FDI issues in dynamic systems and control loops, owing to the application of advanced system and control theory [30]. In this thesis, *we dedicate our effort to model-based FD methods, because the precise mathematical model of the wheeled robot is established via the available mechanical model and system identification.*

The majority of existing research on model-based FDI for MASs focuses on centralized systems, in which all measurements are accessible for detecting and isolating faults in each agent of MASs. For instance, fault detection filters are investigated in [35] concentrating on the problems of communication delay and data loss. In [34], least-square filters and Kalman filters are developed for FDI on time-varying networked sensing systems. Model-based adaptive FD algorithms are used to detect intermittent connections or faults in controller area networks [46, 44]. In general, implementing standard centralized FDI schemes to distributed MASs is extremely challenging for three reasons. At first, not all agents are capable of measuring. Second, distributed MASs are constructed without a powerful centralized entity. Finally, computing power and communication bandwidth are both limited [86]. Using distributed solutions, on the other hand, each agent with a decision-maker can detect faults or cyber-attacks locally, reducing the computation and communication burden dramatically. As a result, *the motivation for this thesis is to investigate distributed FDI strategies, which are more appropriate than centralized ones for MASs.*

Many outstanding results for the application of distributed FDI schemes to MASs have developed in recent years [12, 8]. The distributed FD concept is to design local estimators or FD filters on each intelligent agent based on local sensing and computing resources [30]. In [26], a distributed fault diagnosis architecture for large-scale dynamical systems is proposed based on the overlapping decomposition of the system into sets of interconnected simpler subsystems. Each subsystem is monitored by a fault detector by using local measurements and information from neighboring subsystems. In [27], a similar problem is addressed while the algorithm is extended to a nonlinear uncertainty large-scale discrete

system. However, *for estimation purposes, a portion of the state components must be transferred among multiple nodes via communication, which may cause certain cyber security issues for MASs without physical connections.* In [70], a distributed FDI approach based on unknown input observers (UIOs) in networks of interconnected systems with double integrator dynamics is described, which only requires local measurements. In contrast to [27], the approach in [70] requires each node to measure the state of all its neighbors rather than to communicate with each other, which makes MASs more complicated to implement. [94] shows a sliding model observer-based FDI of distributed networked control systems with time delay, whereas [76] illustrates how the parity space approach can be used in the same situations. To improve the robustness against noises and model uncertainties, recent researches in distributed fault detection and estimation [43, 53] have developed by using adaptive threshold setting methods and sliding-mode algorithms. Each subsystem necessitates not just local input-output measurements, but also measurements of subsystems that are interconnected to and impact the subsystem. Due to the shortcomings of the current state of the art on distributed FDI on MASs, this thesis attempts to address the following issues:

- to improve detection efficiency while lowering communication load, each agent should interact with a sufficient number of agents, including not just its neighbors but also more other agents in the closed appropriate layers, via limited communication iteration.

- instead of state measurement or estimate, innovation signals should be distributed among agents to prevent attackers from identifying the system's parameters and injecting stealthy cyber-attacks against MASs.

- one of the most fundamental difficulties is solving detection and estimation problems using scalable algorithms, which lead to the development of novel distributed algorithms for estimation [61]. A recursive form of computation for updating distributed observers should be explored.

- The distributed FD method should be implemented for online detection.

Why Distributed Detection of Cyber-attacks on MASs?

MASs can be regarded as typical cyber-physical systems (CPSs). With the recent development of large-scale spatial distributed systems and network information technology, the communication networks of MASs are exposed to the public internet, which is vulnerable to being attacked by adversaries [57, 39, 83, 68]. Generally speaking, the cyber attacks of MASs can be categorized into two types: denial-of-service (DoS) attacks

[62, 7, 14] and deception attacks [33, 82, 77]. When MASs are under DoS attacks, the communication links among agents are destroyed or paralyzed by attackers, causing the service of MASs to be temporarily interrupted, stopped, or disrupted. On the other side, for MASs under deception attacks, a typical phenomenon is that some false data are injected into sensor measurements and control signals of actuators, with a great risk of global mission failure, and even damaging the whole system. In this case, the detection of cyber-attacks for MASs has received considerable attention during the last decade.

Early studies on detection of DoS attacks, in [74], a centralized sequential changing-point monitoring algorithm is proposed, based on the inherent network protocol behaviors. And support vector machines technique [58], and neural network methods [2] are also investigated to solve the centralized detection problems of DoS attacks. To author's knowledge, few results have been developed on distributed detection on MASs under DoS attacks. In [9], a distributed change point detection architecture is developed using change aggregation trees to detect abrupt changes in traffic flows based only on transmission data. A similar strategy is proposed in [69], where a distributed algorithm uses the number of users connected to determine whether the agent is linked via leaky buckets at the routers. The observer-based method is described in [13], in which a distributed observer-based attack detector is designed considering model uncertainties and measurement noise. Despite the fact that the misappropriated node can be detected, the proper threshold setting procedure is neglected.

It motivates us to propose a distributed detection scheme for MASs under DoS attacks in this thesis, which fulfills the requirements listed below.

- the distributed detector should be embedded in each agent to determine whether or not the connections between its neighbors have been interrupted.
- a threshold setting algorithm should be investigated to guarantee detection performance.

Concerning the deception attack detection problem for MASs, it should be raised more attention, particularly in the case of consensus cooperative MASs. Adversaries will inject false data into communication channels between agents and modify transmittal signals, such as control signals and measurement data. As a result, maintaining consensual control of MASs under deception attacks is exceedingly challenging.

Some results are focused on centralized detection methods. For instance, in [49], an active synchronous detection method is presented to detect deception attacks on

inverter controllers in micro-grids without impeding system operations, by comparing the specified small probing signals generated by the control center with the output signals of a local controller. Byzantine attacks have already been addressed in the context of decentralized inference [40], in which individual agents make measurements and report them to a fusion center for detection.

On the other hand, investigation of distributed detection methods is becoming a vital requirement for computation reduction and robustness enhancement. for instance, a resilient multi-agent distributed estimation of an unknown vector parameter is studied in [10], when a subset of the agents is adversarial. In [28], authors expand the research of [40] to a distributed form, and aim to detect deception cyber-attacks on a cyber-physical system over a cluster-based network in which several fusion nodes receive data from sensors and collaborate in a neighbor-wise way. Reference [29] offers a distributed attack detection and isolation methodology for DC micro-grids, using a local detection threshold and the locally unknown input observer on each agent to estimate the state of neighbors. Furthermore, The authors even investigate a type of stealthy attack, but no detection strategy is discussed.

Because deception cyber-attacks insert attack signals into the transmission of information via network interfaces, the processes of deception cyber-attacks and integrity attacks are relatively similar. Integrity attacks on remote control systems occur between the controller and actuator, whereas deception cyber-attacks constantly affect data interaction between various agents in MASs. Compared to a few studies on distributed detection of deception cyber-attacks on MASs, integrity attacks detection of remote automatic control systems [15, 32] has more achievements. References [15, 32, 72] address the issues with good detection performance using the observer-based fault detection approach [16]. However, if cyber-attacks are built by adversaries who are familiar with the system and the presence detection mechanism, they are unlikely to be detected. In this instance, the usual observer-based fault detection strategy may be ineffective in detecting stealthy attacks [21].

In this thesis, we investigate how to detect deception cyber-attacks on MASs without relying on a central entity or fusion center, and are focused on studying distributed detection methods. Furthermore, inspired by the research works in [92, 22], and based on a detection scheme with encrypted transmissions of control and monitoring signals in the feedback control system [21], our work in this thesis is motivated to extend the observer-based detection scheme in [21] to the application on distributed detection of deception cyber-attacks for cooperative MASs. At the same time, the detection algorithm

should take into account the following factors.

- to prevent attackers from identifying the system's information, the detection algorithm should send encrypted data between agents instead of input and output data.
- after plugging in the detection system, the performance of MASs should not be affected.

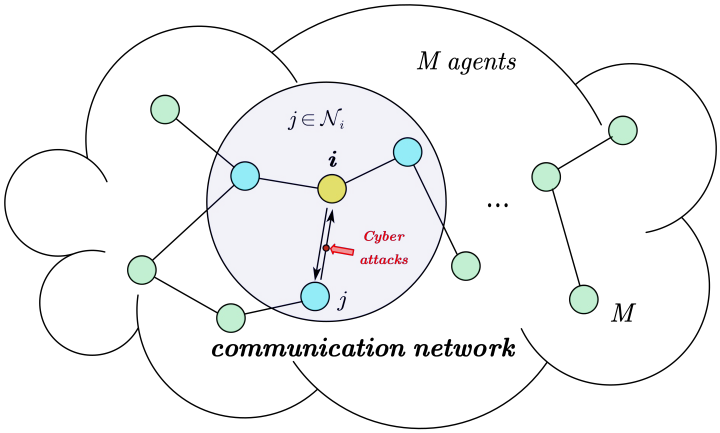

Figure 1.1: Diagram of task description

1.2 Objectives of the Work

Inspired by the aforementioned motivations, the main objective of this work is to develop distributed observer-based detection schemes for faults and cyber-attacks on multi-agent systems. To be specific, as illustrated in Figure 1.1, the tasks of this thesis are stated as follows:

- establish linear time-invariant (LTI) models of a multi-sensors system and a multiple wheeled-robots system, respectively.
- propose a distributed observer-based online fault detection recursive algorithm for detecting abnormal target movement, taking into account a proper threshold setting and a reasonable number of agents for communication iteration.
- develop a distributed online detector integrated in each agent to determine whether or not the communication links between its neighbors are under DoS attack, and set a threshold for each agent to ensure detection performance.

- provide a distributed detection method to solve the deception attack problem without slowing down the system, and construct an encoder-decoder system to encrypt transmitted data to prevent attackers from identifying the dynamics of MASs.

1.3 Outline of the Thesis

This thesis consists of six chapters, which are organized as shown in Fig.1.2. The major objective and contributions of each chapter are briefly summarized as follows.

Chapter 2, "*Preliminaries and Basic Theories*", gives some preliminaries of MASs, concerning graph theory and the definition of multi-layer neighborhood. Then some basics theories including description of dynamic systems, fault detection for LTI systems and fault tolerant control (FTC) configuration are presented. This provides fundamentals for developing new approaches of distributed state estimation and fault detection.

Chapter 3, "*Distributed H_2 observer-based FD for Multi-sensor Networks*", focuses on applying distributed H_2 observer to solve problems of state estimation and small fault detection by using large-scale time-varying sensor networks with high variance of measurement noise. To achieve distributed detection purpose and online implementation, not only the communication models but also the H_2 observer are developed in a recursive form. We also proposed a distributed detection scheme on each sensor, and the performance of estimation and fault detection are verified by simulation case studies. The key contribution is that the distributed H_2 observer can lead to better performance of state estimation and fault detection by using more data transferred from its neighbors for each sensor node.

Chapter 4, "*Distributed Detection of DoS Attacks on MASs*", faces a detection problem against DoS cyber attacks on MASs. Based on the local Kalman filter on each node and a communication iteration method, a generalized likelihood ratio (GLR) method-based online detection algorithm is proposed to handle the detection problem. The corresponding threshold is obtained by an offline statistical training method. At last, we provide a series of straightforward simulation studies on a sensor network to verify the feasibility of the proposed detection algorithm.

Chapter 5, "*Distributed Detection of Deception Attacks on MASs*", deals with problems of distributed detecting deception cyber-attacks distributively on cooperative MASs in a consensus control configuration. In this case, false data added to reference

signals via information transmission could be considered as stealthy cyber-attacks. We propose a distributed detection scheme with an encrypted system for reliable cyber-attack detection without loss of control and monitoring performance. Preventing system dynamics from being obtained by attackers will be also considered. Finally, the effectiveness of the distributed detection algorithm is validated by simulation and experimental studies on a two-robot system.

Chapter 6, "*Conclusion and future works*", finally, provides a summary of the contribution in this thesis and gives a conclusion. Then some future works are discussed in the last part of this chapter.

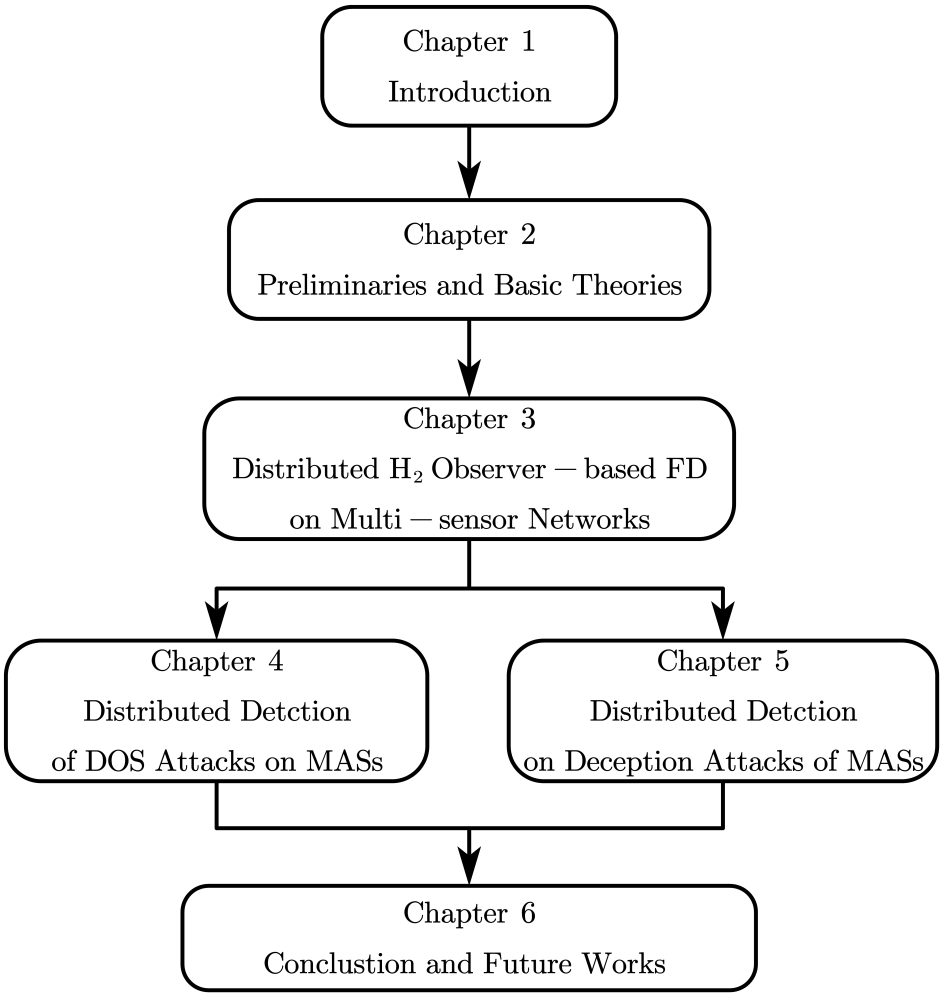

Figure 1.2: Organization of the chapters

2 Preliminaries and Basic Theories

The objective of this chapter is to introduce some preliminaries for our later study. This consists of four parts. The first part concerns some basics of MASs including graph theory and the definition of multi-layer neighborhood. The second part deals with the description of dynamic systems. After that, the basics of fault detection for LTI systems is introduced in the third part of this chapter. Finally, some basics of FTC configuration are presented in the fourth part.

2.1 Basics of MASs

2.1.1 Graph theory

We consider a networked system of M nodes presented by graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \left\{\upsilon_1, \quad \upsilon_2, \quad \cdots \quad \upsilon_M\right\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set of paired nodes in the graph. Hereafter in this thesis, we simplify the description of the i^{th} node from node υ_i to node i. In a directed graph, $(i, j) \in \mathcal{E}$ means data can be transmitted from node i to node j, but not the other way around. However, $(i, j) \in \mathcal{E}$ is equivalent to $(j, i) \in \mathcal{E}$ in an undirected graph.

Definition 2.1. *A connected graph is defined as an undirected graph with a path connecting every pair of nodes; otherwise, it is referred to as a disconnected graph.*

As illustrated in Figure 2.1, we only investigate connected undirected graphs with wireless communication networks in our study.

Definition 2.2. *In an undirected graph, two nodes i and j are defined as neighbors if $(i, j) \in \mathcal{E}$ and they can communicate directly with each other.*

The neighborhood set of node i is defined by $\mathcal{N}_i = \{j \in \mathcal{V}, (i, j) \in \mathcal{E}\}$. For instance, the node $j \in \mathcal{N}_i$, shown in blue in Figure 2.1, is the neighbor of node i.

$\mathcal{A} = [a_{ij}] \in \mathcal{R}^{M\times M}$ is the adjacent matrix of graph $\mathcal{G}$, here a_{ij} indicates an element of $\mathcal{A}$ and defined by

$$a_{ij} = \begin{cases} 1 & i \neq j \ and \ j \in \mathcal{N}_i \\ 0 & otherwise \end{cases} \tag{2.1}$$

We define the degree matrix as $\mathcal{D} = \text{diag}\,(d_{g,1}, d_{g,2}, \cdots, d_{g,M}) \in \mathcal{R}^{M\times M}$, with $d_{g,i} = \sum_{j=1}^{M} a_{ij}$ denoting the degree of the node i. The value of $d_{g,i}$ equals the number of edges linked to node i, as well as the number of neighbors for node i.

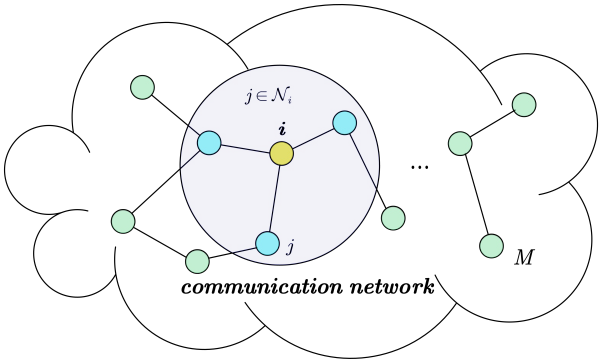

Figure 2.1: Graph of a networked system

Based on the adjacent matrix $\mathcal{A}$ and degree matrix $\mathcal{D}$, the laplacian matrix $\mathcal{L} = [\mathcal{L}_{ij}] \in \mathcal{R}^{M\times M}$ of graph $\mathcal{G}$ is given as follows:

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \tag{2.2}$$

The eigenvalues of the laplacian matrix $\mathcal{L}$ are denoted by $\lambda_i \in \mathcal{R}$, $i = 1, \cdots, M$. In a connected graph, $\mathcal{L}$ has just one zero eigenvalue and all nonzero eigenvalues are all positive and on the open right half-plane [31]. In this case, all of the eigenvalues of $\mathcal{L}$ can be stated as follows in ascending order of magnitude:

$$0 = \lambda_1 < \lambda_2 \leqslant \cdots \leqslant \lambda_M$$

where λ_1 equals to zeros and represents the smallest eigenvalue of $\mathcal{L}$, while λ_M indicates the largest eigenvalue.

2.1.2 Multi-layer neighborhood

In a communication network, each node could transfer its information not only to its neighbors directly but also to the other nodes in a graph indirectly via communication

nodes by nodes. It motivates us to extend the definition of the neighborhood from the one in Subsection (2.1.1) to a multi-layer-neighborhood.

Before we discuss the concept of multi-layer neighborhood, let $g(i,j)$ be the distance between the node i and node j, $i,j \in \mathcal{V}$, which represents the minimum length of the paths connecting the two nodes. Take node i as an example, which is shown in Figure 2.2. The distance from node i to itself is defined as $g(i,i) = 0$. According to the definition of the neighborhood in Subsection (2.1.1), if $j \in \mathcal{N}_i$, then it is clear that $g(i,j) = 1$. In Figure 2.2, if we define the distance between node i and node j equals ξ, which means ξ is the minimal communication iterations for transmission data between the two nodes. Consequently, we can extend the definition of neighborhood to the nodes among ξ-steps communication iterations with node i as ξ-layers neighborhood for the i^{th} node.

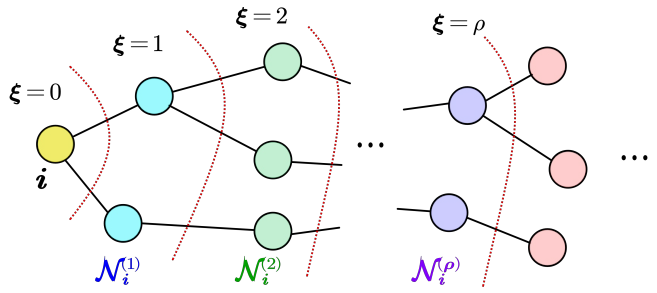

Figure 2.2: Sketch map of multi-layer neighborhood

Definition 2.3. *A ξ-layers neighborhood of node i in a connected graph is made up of nodes whose distance from node i is less than or equal to ξ. And ρ-layers is the greatest extent of our interest in this study.*

$$\mathcal{N}_i^{(\xi)} = \begin{cases} i & \xi = 0 \\ j & \xi > 0 \text{ and } \xi \leqslant \rho \end{cases} \tag{2.3}$$

This concept will aid our later study in Chapters 3 and 4 on the construction of a distributed observer and determining the detection range of cyber-attacks, respectively.

2.2 Modeling of Dynamic Systems

In this section, we devote ourselves to introducing models for the mathematical description of dynamic systems. To analyze the system behavior in both fault-free and faulty cases, we first provide a brief overview of different forms of models, including, but not limited to, i) input-output description, ii) state-space representation, and iii) models with disturbances

and various types of faults. Moreover, coprime factorization technique, as one of the fundamental tools for our research, provides a further system representation form, which will be utilized later in this thesis.

2.2.1 Description of nominal systems

A dynamic system can be represented in a variety of forms depending on the goal of modeling. The simplest form is the so-called linear time-invariant (LTI) system model, which is widely utilized in research and application areas. LTI systems in the situation of disturbance-free and fault-free are called as nominal systems. Here, we briefly introduce the following two standard mathematical model forms for LTI systems:

- transfer matrix,
- state-space representation.

Transfer matrix

A transfer matrix is an input-output description of the dynamic behavior of an LTI system in the frequency domain. Let a nominal plant model $G(z)$ represent a transfer matrix from the input vector $u \in \mathcal{R}^{n_u}$ to the output vector $y \in \mathcal{R}^{n_y}$, that is,

$$y(z) = G(z)\, u(z) \tag{2.4}$$

We assume that $G(z)$ is a proper real-rational matrix. Here z denotes the complex variable of z-transform for discrete-time signals. In this thesis, we primarily focus on discrete-time processes.

State-space representation

The minimal state-space representation of the nominal discrete LTI system G is given by

$$G : \begin{cases} x(k+1) = Ax(k) + Bu(k), x(0) = x_0 \\ y(k) = Cx(k) + Du(k) \end{cases} \tag{2.5}$$

where $k = \{0, \quad 1, \quad 2, \quad \cdots\}$ is sampling instants, $x(k) \in \mathcal{R}^{n_x}$ denotes the state vector and $x(0) = x_0$ is the initial state of the plant G. A, B, C and D are real constant matrices with appropriate dimensions. Due to the assumption of minimal realization, (A, B) is controllable, and (A, C) is observable. For simplicity, the corresponding transfer matrix

for discrete system Eq.2.5 can be written as

$$G(z) = C(zI - A)^{-1} B + D \tag{2.6}$$

which is also denoted by

$$G(z) = (A, B, C, D) \tag{2.7}$$

Notably, we assume that the (A, B, C, D) is the minimal realization of G. For simplistic, later we use the following block notation to present a transfer matrix

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = C(zI - A)^{-1} B + D \tag{2.8}$$

If the process Eq.2.5 is corrupted by stochastic disturbances, such as process noise $\omega(k)$ and measurement noises $\nu(k)$, the corresponding state-space representation is extended as

$$G: \begin{cases} x(k+1) = Ax(k) + Bu(k) + \omega(k), x(0) = x_0 \\ y(k) = Cx(k) + Du(k) + \nu(k) \end{cases} \tag{2.9}$$

We assume the process noise $\omega(k)$ and measurement noise $\nu(k)$ to be zero-mean, normally distributed Gaussian white processes with the known variance matrices Σ_ω and Σ_ν, and be independent of the input vector $u(k)$ and initial state vector $x(0)$.

$$\mathcal{E}\left(\begin{bmatrix} \omega(\xi) \\ \nu(\xi) \\ x(0) \end{bmatrix} \begin{bmatrix} \omega(\zeta) \\ \nu(\zeta) \\ x(0) \end{bmatrix}^T\right) = \begin{bmatrix} \begin{bmatrix} \Sigma_\omega & S \\ S^T & \Sigma_\nu \end{bmatrix} \delta_{\xi\zeta} & 0 \\ 0 & \Pi_0 \end{bmatrix}, \delta_{\xi\zeta} = \begin{cases} 1, \xi = \zeta, \\ 0, \xi \neq \zeta \end{cases} \tag{2.10}$$

2.2.2 Description of systems with disturbances and faults

Based on linear FD architecture [16], the systems G_f considering disturbances and faults, as shown in Figure 2.3, are modeled by extending Eq.2.9 to

$$G_f: \begin{cases} x(k+1) = Ax(k) + Bu(k) + E_d d(k) + E_f f(k) + \omega(k), x(0) = x_0 \\ y(k) = Cx(k) + Du(k) + F_d d(k) + F_f f(k) + \nu(k) \end{cases} \tag{2.11}$$

where $d(k) \in \mathcal{R}^{k_d}$ represents unknown disturbance vectors. It is assumed that $d(k)$ is l_2 bounded with

$$\|d\|_2^2 \leqslant \delta_d^2. \tag{2.12}$$

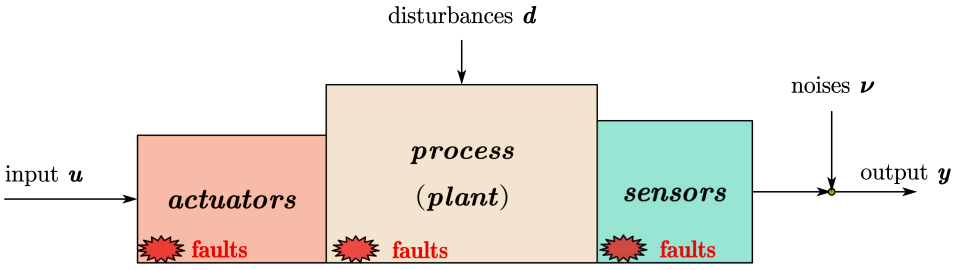

Figure 2.3: Schematic description of the processes with disturbances and faults [16]

And $f \in \mathcal{R}^{k_f}$ is an unknown vector that represents all possible faults to be detected, which are illustrated in Figure 2.3. E_d, E_f, F_d and F_f are known matrices of appropriate dimensions. Based on the location of faulty cases, faults can be divided into actuator faults, process faults and sensor faults [17], which can be, respectively, described as

- actuator faults:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E_d d(k) + Bf(k) + \omega(k), x(0) = x_0 \\ y(k) = Cx(k) + Du(k) + F_d d(k) + Df(k) + \nu(k) \end{cases} \tag{2.13}$$

- process faults:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E_d d(k) + E_p f(k) + \omega(k), x(0) = x_0 \\ y(k) = Cx(k) + Du(k) + F_d d(k) + F_p f(k) + \nu(k) \end{cases} \tag{2.14}$$

  where E_p and F_p are the matrices related to the plant

- sensor faults:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E_d d(k) + \omega(k), x(0) = x_0 \\ y(k) = Cx(k) + Du(k) + F_d d(k) + f(k) + \nu(k) \end{cases} \tag{2.15}$$

$f(k)$ is assumed to be a deterministic time function, and it will be zero in the fault-free

case. It can be described as

$$\begin{cases} f(k) = 0, & k < k_f, fault - free \\ f(k) \neq 0, & k \geqslant k_f, faulty \end{cases} \tag{2.16}$$

We categorize faults into two types based on how they influence the dynamics of the system:

- additive faults: the faults described by Eq.2.11 have no effect on the system's stability and are independent by the configuration.

- multiplicative faults: the faults have the ability to change system dynamics, which are presented by extending Eq.2.9 to

$$\begin{cases} x(k+1) = (A + \Delta A_f) x(k) + (B + \Delta B_f) u(k) + \omega(k), x(0) = x_0 \\ y(k) = (C + \Delta C_f) x(k) + (D + \Delta D_f) u(k) + \nu(k) \end{cases} \tag{2.17}$$

where ΔA_f, ΔB_f, ΔC_f and ΔD_f denote the multiplicative faults in the plant.

The coprime factorization techniques will be introduced next, which are essential for formulating the FD and FTC configurations of LTI systems.

2.2.3 Coprime factorization techniques

LTI systems can be represented in a different way using coprime factorization of a transfer function. Generally speaking, a transfer matrix can be factored as the product of two stable and coprime transfer matrices.

Definition 2.4. *[21] Given two transfer matrices $M(z) \in \mathcal{RH}_\infty$ and $N(z) \in \mathcal{RH}_\infty$ are called right coprime over $\mathcal{RH}_\infty$, if there exist another two transfer matrices $X(z) \in \mathcal{RH}_\infty$ and $Y(z) \in \mathcal{RH}_\infty$ such that*

$$\begin{bmatrix} X(z) & Y(z) \end{bmatrix} \begin{bmatrix} M(z) \\ N(z) \end{bmatrix} = I \tag{2.18}$$

Definition 2.5. *[21] Given two transfer matrices $\hat{M}(z) \in \mathcal{RH}_\infty$ and $\hat{N}(z) \in \mathcal{RH}_\infty$ are defined as left coprime over $\mathcal{RH}_\infty$, if there exist two transfer matrices $\hat{X}(z) \in \mathcal{RH}_\infty$ and*

$\hat{Y}(z) \in \mathcal{RH}_\infty$ *such that*

$$\begin{bmatrix} \hat{M}(z) & \hat{N}(z) \end{bmatrix} \begin{bmatrix} \hat{X}(z) \\ \hat{Y}(z) \end{bmatrix} = I \tag{2.19}$$

Definition 2.6. *A factorization $G(z) = N(z)M^{-1}(z)$ is called to be an right-coprime factorization (RCF) [45] of $G(z)$, while $G(z) = \hat{M}^{-1}(z)\hat{N}(z)$ with $\hat{M}(z) \in \mathcal{RH}_\infty$ and $\hat{N}(z) \in \mathcal{RH}_\infty$ is said to be an left-coprime factorization (LCF) of $G(z)$.*

According to [16], the LCF of $G(z)$ is essential to design the so-called residual generator. And the RCF will play a key role in designing cyber-attack detector in later chapters.

Lemma 1. *[16] Assuming the plant G (Eq.2.5) is a proper real-rational transfer matrix, and it is controllable(stabilizable) and observable(detectable). Choosing a state feedback gain matrix F and observer gain matrix L ensuring $A + BF$ and $A - LC$ are Schur matrices, and*

$$\hat{M}(z) = (A - LC, -L, C, I), \qquad \hat{N}(z) = (A - LC, B - LD, C, D) \tag{2.20}$$

$$M(z) = (A + BF, B, F, I), \qquad N(z) = (A + BF, B, C + DF, D) \tag{2.21}$$

$$\hat{X}(z) = (A + BF, L, C + DF, I), \qquad \hat{Y}(z) = (A + BF, -L, F, 0) \tag{2.22}$$

$$X(z) = (A - LC, -(B - LD), F, I), \; Y(z) = (A - LC, -L, F, 0) \tag{2.23}$$

and

$$\begin{bmatrix} X(z) & Y(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix} = \left[\begin{array}{c|cc} A - LC & -(B - LD) & -L \\ \hline F & I & 0 \\ C & -D & I \end{array}\right] \tag{2.24}$$

$$\begin{bmatrix} M(z) & -\hat{Y}(z) \\ N(z) & \hat{X}(z) \end{bmatrix} = \left[\begin{array}{c|cc} A + BF & B & L \\ \hline F & I & 0 \\ C + DF & D & I \end{array}\right] \tag{2.25}$$

Moreover, the so-called Bezout identity [45] holds

$$\begin{bmatrix} X(z) & Y(z) \\ -\hat{N}(z) & \hat{M}(z) \end{bmatrix} \begin{bmatrix} M(z) & -\hat{Y}(z) \\ N(z) & \hat{X}(z) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{2.26}$$

2.3 Basics of Fault Detection for LTI Systems

A standard observer-based FD system, as shown in Fig.2.4, consists of an observer-based residual generator, an evaluator, a threshold setting block and a decision logic.

2.3.1 Observer-based residual generator

Fault detection filter

The fault detection filter (FDF) is an observer-based residual generator that is originally proposed by Beard [6] in the early 1970s. According to the nominal model (Eq.2.5), we consider a full order state observer below, which is the most widely used residual generator.

$$\hat{G}: \begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)) \\ \hat{y}(k) = C\hat{x}(k) + Du(k) \end{cases} \tag{2.27}$$

where $\hat{x}$ and $\hat{y}$ denote the estimation of the state vector and output vector, respectively. The observer gain matrix, represented by L, is meant to meet the requirement that $A-LC$ is a Schur matrix. Built upon Eq.2.27, the residual signal is simply defined as

$$r_0(k) = y(k) - \hat{y}(k) = y(k) - Cx(k) - Du(k) \tag{2.28}$$

The residual vector $r_0(k)$ indicates the change in the system caused by the unknown input vectors, such as faults, disturbances, and noises, if $y(k)$ is generated by the system G_f described in Eq.2.11.

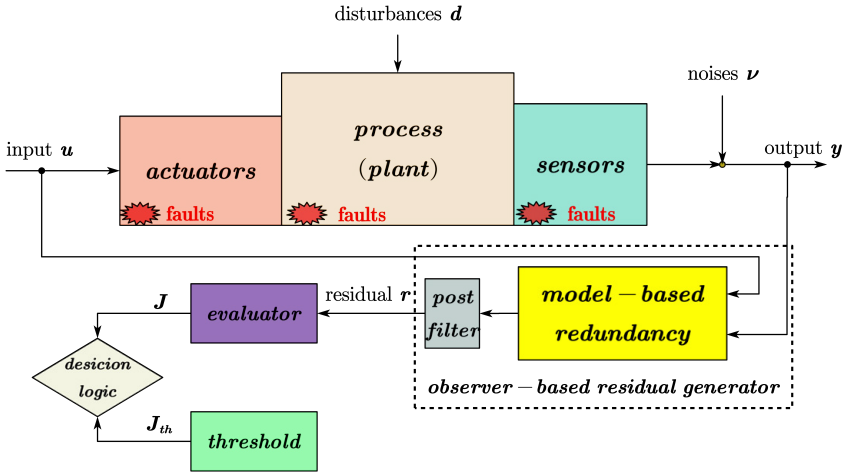

Figure 2.4: Observer-based fault detection configuration [17]

Considering the vector of estimation error $e(k) = x(k) - \hat{x}(k)$, it yields

$$e\left(k+1\right) = \left(A - LC\right) e\left(k\right) \tag{2.29}$$

$$r_0\left(k\right) = Ce\left(k\right) \tag{2.30}$$

Because $A - LC$ matrix is stable, the residual will converge to zero for any initial state $x(0)$ and input signal of plant $u(k)$. It satisfies

$$\forall u\left(k\right), x\left(0\right), \lim_{k\to\infty} r\left(k\right) = 0 \tag{2.31}$$

In this case, $\hat{x}$ provides an unbiased estimation for state vector x, and it exits

$$\lim_{k\to\infty} \left(x\left(k\right) - \hat{x}\left(k\right)\right) = 0 \tag{2.32}$$

Based on Eq.2.27 and Eq.2.28 and LCF $G\left(z\right) = \hat{M}^{-1}\left(z\right)\hat{N}\left(z\right)$, it is worth noting that a residual generator can be formulated by

$$r_0\left(z\right) = \hat{M}\left(z\right)y\left(z\right) - \hat{N}\left(z\right)u\left(z\right) = \begin{bmatrix} -\hat{N}\left(z\right) & \hat{M}\left(z\right) \end{bmatrix} \begin{bmatrix} u\left(z\right) \\ y\left(z\right) \end{bmatrix} \tag{2.33}$$

From Eq.2.33 we can see, the input vectors are u, y while the residual signal vector r_0 is the output. In [16], the dynamic system $\begin{bmatrix} -\hat{N}\left(z\right) & \hat{M}\left(z\right) \end{bmatrix}$ is called kernel representation of nominal system (Eq.2.5).

Definition 2.7. *Given the system model (Eq.2.5) and a corresponding LCP* $\left(\hat{M}, \hat{N}\right)$*, we define the following representation*

$$\mathcal{K}_p = \left\{ \begin{bmatrix} u \\ y \end{bmatrix} : \begin{bmatrix} -\hat{N} & \hat{M} \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0, \begin{bmatrix} u \\ y \end{bmatrix} \in \mathcal{H}_2 \right\} \tag{2.34}$$

which represents the kernel subspace of $\mathcal{H}_2 \times \mathcal{H}_2$*. The kernel space is a closed subspace, in which all (bounded) input and output pairs* (u, y) *satisfy [73],*

$$\begin{bmatrix} -\hat{N}\left(z\right) & \hat{M}\left(z\right) \end{bmatrix} \begin{bmatrix} u\left(z\right) \\ y\left(z\right) \end{bmatrix} = 0 \tag{2.35}$$

According to the proof in [23], any LTI residual generator can be parameterized by

$$r\left(z\right) = R\left(z\right)\left(\hat{M}\left(z\right)y\left(z\right) - \hat{N}\left(z\right)u\left(z\right)\right) \tag{2.36}$$

here $R(z)$ denotes the stable parameterization transfer function matrix, i.e. the post-filter. By increasing the degree of design freedom, $R(z)$ is meant to achieve high sensitivity to faults and robustness against disturbances.

Kalman filter based residual generator

Based on the fundamental papers by Kalman and Bucy [42, 41] in the 1960s, the classical Kalman filter (KF) has been used in a variety of domains, including control systems and wireless communications. A linear state-space model represents the system's dynamics and observations, containing the system process and measurement noises that affect the state predictions and estimations.

We consider a discrete-time LTI system with process and measurement noise vectors as given by Eq.2.9. The noises are supposed to satisfy the condition Eq.2.10. As a result, a Kalman filter generates residual vector which is a white Gaussian process.

Due to the following recursions of computation, a Kalman filter is a time-varying system, which is summarized step by step in Algorithm 1.

where $\hat{x}(k)$ denotes the state estimation. To obtain the optimal state estimation, $P(k)$ represents the covariance of state estimation error $e(k) = x(k) - \hat{x}(k)$ which should fulfill the requirement by

$$P(k) = \mathcal{E}\left[e(k)e^T(k)\right] = min \tag{2.37}$$

The residual signal provided by a Kalman filter should be a zero-mean white Gaussian process in fault-free conditions. When systems suffer faults or cyber-attacks, changes in the mean value, the covariance of the residual, or both can be detected by using, for example, a generalized likelihood ratio (GLR) test, which will be detailed in Chapter 4.

2.3.2 Residual evaluation and threshold setting based on statistical method

We are now in a position to investigate how to evaluate residual signals and provide a suitable threshold. Because the residuals are often coupled with disturbances and uncertainties, the evaluation function of the residual signal helps us to distinguish faults from disturbances and uncertainties. A simple comparison between the evaluation function and the threshold will then be used to make a decision on the possibility of a fault, as

illustrated in Figure 2.4.

Algorithm 1 Online recursive computation of Kalman filter

Step 1: Initial conditions when $k = 0$

$$\hat{x}(0) = 0, P(0) = \Pi_0 \tag{2.38}$$

Step 2: Residual signal generation

$$r_{0K}(k) = y(k) - C\hat{x}(k) - Du(k) \tag{2.39}$$

Step 3: Covariance of residual computation

$$\mathcal{E}(r_{0K}(k)r_{0K}^T(k)) = R_e(k) = \Sigma_\nu + CP(k)C^T \tag{2.40}$$

Step 4: Time-varying Kalman filter gain obtain

$$L_K(k) = \left(AP(k)C^T + S\right) R_e^{-1}(k) \tag{2.41}$$

Step 5: State estimation updating

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L_K(k)r_{0K}(k) \tag{2.42}$$

Step 6: Covariance of estimation error updating

$$P(k+1) = AP(k)A^T - L_K(k)R_e(k)L_K^T(k) + \Sigma_\omega \tag{2.43}$$

Step 7: $k = k + 1$ and repeat step 2 to step 6

Basic hypothesis test and elementary concepts

To clarify some basic concepts of fault detection, we consider a simple fault detection problem. Given a measurement model

$$y = \theta + \nu \in \mathcal{R} \tag{2.44}$$

with $\nu \sim N(0, \Sigma_\nu)$ denoting the measurement noise under normal distribution with zero mean and known variance Σ_ν. And $\theta = 0$ represents the fault-free case, while $\theta \neq 0$ represents the faulty case. Our detection problem here is expressed as, how can we select a threshold J_{th} for reliable detection of the fault (change) in θ by using a series of samples of the measurement y?

By hypothesis testing, with the aid of null hypothesis H_0 and alternative hypoth-

esis H_1, the above detection problem can be formulated as:

$$\begin{cases} H_0, & \text{nullhypothesis}: & \theta = 0, & fault - free \\ H_1, & alternative\ hypothesis: & \theta \neq 0, & faulty \end{cases} \tag{2.45}$$

A function of the samples will be specified as a random variable to make a trustworthy decision about determining faults whether the test statistic supports the alternative hypothesis H_1 or rejects the null hypothesis H_0. We shall define a test statistic by J having the following properties in our later study:

$$\begin{cases} J \in [0, J_{th}], & H_0 & fault - free \\ J \in (J_{th}, \infty), & H_1 & faulty \end{cases} \tag{2.46}$$

The above description can also be regarded as the fault detection logic.

Because the test statistic J is a random variable, the decision logic may raise an alarm even though no fault has happened. In this case, we can define that

Definition 2.8.

$$\text{prob}\{J > J_{th} | \theta = 0\} = \alpha > 0 \tag{2.47}$$

here, the probability α is called significance level, or false alarm rate (FAR), which means the probability of a (wrong) decision in fault-free cases.

Besides, let me introduce other two fundamental definitions that are commonly used to evaluate FD performance.

Definition 2.9. *we define missed detection rate (MDR) as*

$$\text{prob}\{J \leq J_{th} | \theta \neq 0\} \tag{2.48}$$

which is the probability that the decision alarm is missed triggered when faults or changes are present (H_1).

Definition 2.10.

$$\text{prob}\{J > J_{th} | \theta \neq 0\} \tag{2.49}$$

can be defined as fault detection rate (FDR), which indicates the probability of being detected when a fault occurs (H_1).

Based on the above three definitions, the threshold J_{th} should be selected to fulfill the requirements as follows:

- with the threshold, the FAR equals to an admissible rate
- meanwhile, the MDR is desired to be minimized and FDR should be maximized, due to

$$FDR = 1 - MDR \tag{2.50}$$

2.4 Basics of FTC Configuration

Consider the standard feedback control loop, as shown in Figure 2.5

$$\begin{cases} y(z) = G(z)\, u(z) \\ u(z) = K(z)\, y(z) \end{cases} \tag{2.51}$$

with the plant model G in a state-space form represented in Eq.2.5, and $K(z)$ denotes the transfer function of a feedback controller.

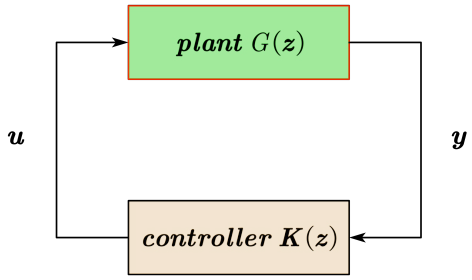

Figure 2.5: Sketch of standard feedback control loop

As proved in [91], all stabilizing controllers $K(z)$ can be left and right coprime parameterized by

$$K(z) = -\left(X(z) - Q(z)\hat{N}(z)\right)^{-1}\left(Y(z) + Q(z)\hat{M}(z)\right) \tag{2.52}$$

$$= -\left(\hat{Y}(z) + M(z)Q(z)\right)\left(\hat{X}(z) - N(z)Q(z)\right)^{-1} \tag{2.53}$$

with $Q(z) \in \mathcal{RH}_\infty$ denoting the Youla parameter system. The parameterization expression Eq.2.52- Eq.2.53 is called Youla parameterization. And according to the four coprime pairs $(\hat{M}, \hat{N})$, (M, N), $(\hat{X}, \hat{Y})$ and (X, Y), it is demonstrated by [20] that any (stabling)

out feedback controller

$$u(z) = K(z)\, y(z) + v(z) \tag{2.54}$$

with $v(z)$ representing the reference signal could be formulated equivalently as

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Lr_0(k) \tag{2.55}$$

$$= (A - LC)\,\hat{x}(k) + (B - LD)\,u(k) + Ly(k) \tag{2.56}$$

$$r_0(k) = y(k) - C\hat{x}(k) - Du(k) \tag{2.57}$$

$$u(z) = F\hat{x}(z) - Q(z)\, r_0(z) + \bar{v}(z) \tag{2.58}$$

$$\bar{v}(z) = \left(X(z) - Q(z)\,\hat{N}(z)\right) v(z) \tag{2.59}$$

Using the controller parameterization, we can have another interpretation that any output feedback controller is an observer-based controller driven through the residual vector $r_0(k)$.

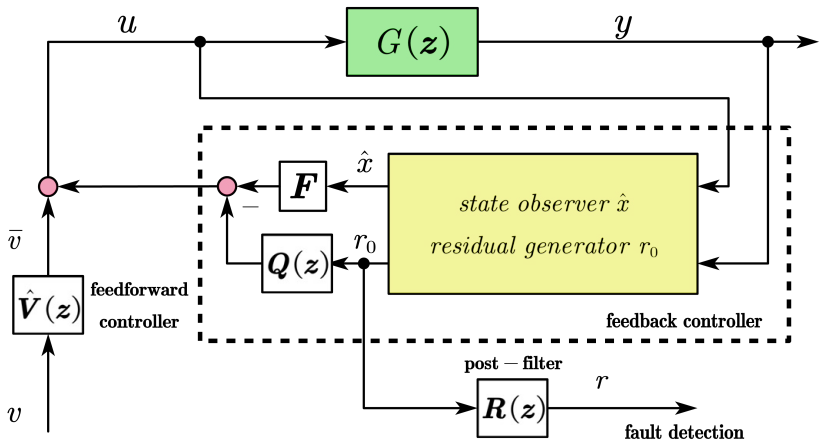

Figure 2.6: Sketch of fault tolerant control architecture

Combined with an observer-based residual generator and a post-filter as we mentioned in Subsection 2.3.1, the well-known fault tolerant control architecture, as shown in Figure 2.6, is composed of three modules:

- an observer based residual generator, which is represented by Eq.2.55-Eq.2.57, provides state estimation and residual signal for the controller and diagnostic system.
- controllers are formulated by Eq.2.58, including an observer-based state feedback controller $F\hat{x}(z) - Q(z)\, r_0(z)$ and a feed-forward controller $\left(X(z) - Q(z)\,\hat{N}(z)\right) v(z)$.

- diagnostic residual generator $R(z)r_0(z)$ for the purpose of fault diagnosis.

In our subsequent study, the fault-tolerant control scheme will be applied on consensus control for MASs to reject disturbances. The concept [17] that the control signal $u(k)$ can be interpreted as an estimate for $Fx(k) + \bar{v}(k)$ will also be addressed as a core of distributed detection of deception cyber-attack for MASs.

2.5 Concluding Remarks

This chapter gives a brief introduction of some preliminaries. This consists of four main aspects. One aspect is the definition of graph theory and multi-layer neighborhood, which are essentials for later description of MASs. The second part deals with the description of dynamic systems, considering disturbance and fault. Using coprime factorization techniques of transfer functions, we can represent LTI systems in a different way. The third part deals with some basics of fault detection for LTI systems. a standard observer-based FD system has been introduced, including an observer-based residual generator, an evaluator, a threshold-setting block, and a decision logic. In the last aspect of this chapter, We deal with the basics of FTC configuration. Using the controller parameterization, we can have another interpretation that any output feedback controller is an observer-based controller driven through the residual vector, which is also a core of distributed detection of deception cyber-attack for MASs.

3 Distributed H_2 Observer-based Fault Detection of Multi-sensor Networks

In many applications involving large-scale complex systems (such as power grid, transportation systems, industrial plants, etc.), the system state is monitored by a group of sensors spatially distributed over large sparse networks. In this regard, we would like to achieve a better estimation performance for each sensor node by using measurement data from its neighbors. To be specific, this chapter attempts to address the following issues

- When designing the observer scheme, it is essential to ensure scalability. This means the scheme should be flexible enough to adjust when nodes are added or removed from the original sensor network. To achieve this adaptability, we need to create an online implementation. In this context, it is worth exploring a recursive form of the distributed H_2 observer.

- Although more data will lead to better estimation performance, the communication iteration should be limited to avoid increasing the communication load.

- Even when dealing with low-quality sensors that exhibit significant measurement noise variance, we still tend to do fault detection on each sensor by using more data through communication iteration with other nodes.

In this chapter, first, we study the modelling of system and sensor-network, and a distributed fault detection problem is discussed. Then, a communication model is designed in a recursive form. After that, we design a distributed H_2 observer also in a recursive form, which can be implemented step by step. Furthermore, a distributed detection scheme realized on each sensor is proposed. Finally, the performance of estimation and fault detection are verified by a simulation case study.

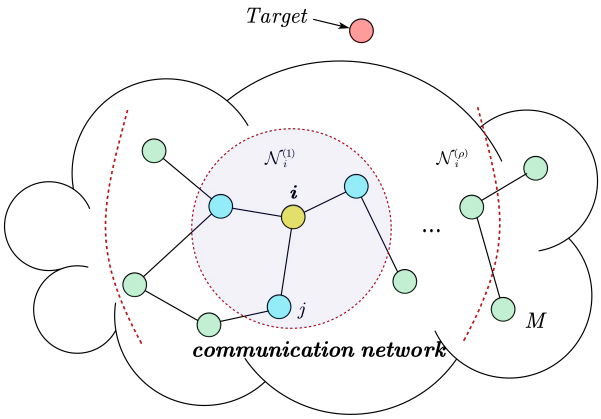

Figure 3.1: Sketch map of sensor network with target

3.1 Problem Formulation

To model the scenario as shown in Fig.3.1 , we describe the minimal state space representation of a nominal discrete linear time-invariant (LTI) dynamical target plant as (3.1), considering unknown input vectors.

$$x\left(k+1\right) = Ax\left(k\right) + d(k) \tag{3.1}$$

where k is the discrete-time index. $x(k) \in \mathcal{R}^n$ denotes the state vector and $x(0) = x_0$ is the initial condition of the target plant. $A \in \mathcal{R}^{n\times n}$ denotes the system matrix, and $d(k) \in \mathcal{R}^n$ represents unknown input vector. It is assumed that $d(k)$ is l_2 bounded with

$$\|d\|_2^2 \leqslant \delta_d^2 \tag{3.2}$$

A network of M sensors are deployed to monitor the target state (3.1). The corresponding graph is undirected and connected. Each of the sensors receives a measurement of the target state at every time step. Specifically, the i^{th} sensor provides the local measurement y_i, given by

$$y_i(k) = Cx(k) + \nu_i(k) \in \mathcal{R}^{m_i},\ i = 1, \cdots, M \tag{3.3}$$

where $C \in \mathcal{R}^{m\times n}$ denotes the output matrix of all the agent. $\nu_i(k) \in \mathcal{R}^{m_i}$ is the measurement noise vector of sensor i, which is a zero mean white noise with variance Σ_{ν_i}, and is assumed to be uncorrelated with the state $x(k)$ and the measurement of other

nodes.

$$\mathcal{E}\left(\nu_i \nu_j^T\right) = \begin{cases} \Sigma_{\nu_i}, & i = j \\ 0, & i \neq j \end{cases} \quad i = 1, \cdots, M \tag{3.4}$$

Assumption 3.1. *Here we assume that the sampling rate and the output matrix at different sensor nodes could be the same.*

To simplify our subsequent work, it is assumed that

$$T_{s,i} = T_s, \ i = 1, \cdots, M \tag{3.5}$$

Here, $T_{s,i}$ is the sampling time of the i^{th} sensor, while T_s denotes the sampling time of the target plant. In order to model the physical faults on the target plant, the dynamic model (3.1) is extended to

$$x\left(k+1\right) = Ax\left(k\right) + d(k) + f(k) \tag{3.6}$$

with $f(k) \in \mathbb{R}^n$ denoting the fault vector, e.g. actuator faults, which is described as

$$\begin{cases} f\left(k\right) = 0, & kT_s < t_f, fault - free \\ f\left(k\right) \neq 0, & kT_s \geqslant t_f, faulty \end{cases} \tag{3.7}$$

where t_f represents the time instant when the fault f appears on the target plant. In this regards, the influence for the measurement on each sensor node could be modeled as

$$\begin{cases} x\left(k+1\right) = Ax\left(k\right) + d\left(k\right) + f\left(k\right), x\left(0\right) = x_0, \\ y_i\left(k\right) = Cx\left(k\right) + \nu_i\left(k\right) \end{cases} \tag{3.8}$$

In the subsequent work, a distributed H_2 observer will be designed for estimation of target state and the purpose of fault detection, using the local measurement and the data received from other nodes.

3.2 Communication Modeling in a Recursive Form

3.2.1 An extended definition for neighborhood

We model a sensor network composed of $M(k)$ nodes as a time-varying graph $\mathcal{G} = \{V, E\}$. V is the set of vertices representing the sensors. $E \subseteq V \times V$ is the set of edges. $(i, j) \in E$

if and only if sensor i and j can communicate directly with each other. The neighborhood of sensor i is defined as $\mathcal{N}_i(k) = \{j \in V : (i,j) \in E\}$. Here, we extend the definition of general neighborhood to ρ-steps neighborhood, $\mathcal{N}_i^{(\rho)}(k)$. ρ represents the maximum number of links between node i and the nodes in the extend neighborhood . An alternative interpretation is that ρ represents the maximum communication iteration of transferring information from the node to node i.

According to the definition, $\mathcal{N}_i^{(0)}(k)$ represents only node i itself. When ρ equals 0, which means only the local information of node i is available. And the conventional definition of neighbor could be defined as $\mathcal{N}_i^{(1)}(k)$. To this end, the ρ-steps neighborhood of node i is represented in a recursive form.

$$\mathcal{N}_i^{(\rho)}(k) = \begin{cases} \{i\} & \rho = 0 \\ \mathcal{N}_i(k) \bigcup \{i\} & \rho = 1 \\ \bigcup_{j \in \mathcal{N}_i(k)} \mathcal{N}_j^{(\rho-1)}(k) & \rho \geq 2 \end{cases} \tag{3.9}$$

Assumption 3.2. *Between the time instant k and $k+1$, we assume the state of the process x is constant.*

Assumption 3.3. *The variance matrix of measurement noise Σ_{v_i} and communication noise Σ_{w_i} at each node i are known and time invariant.*

The system monitoring at each node i includes local measurement, $y_i(k)$, and the information from its ρ-steps neighborhood, $y_{i,\xi}(k)$, $\xi = 0, 1, \cdots, \rho$. As shown in Fig.3.2, we define that ξ is the instant of communication iteration, where ρ is the maximum iteration number depending on the communication degree of the sensor networks.

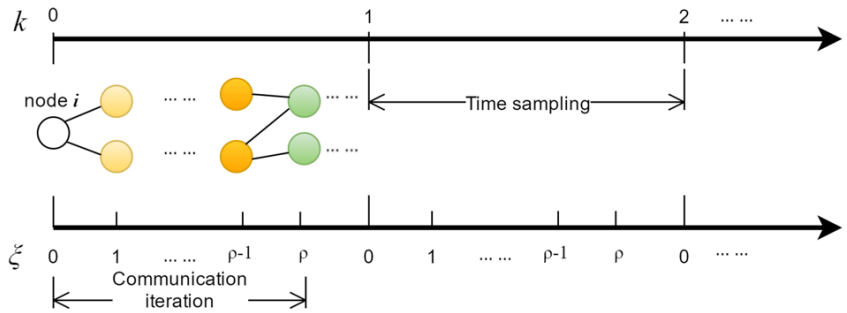

Figure 3.2: Communication iteration between sampling time of target

Assumption 3.4. *For the simplicity issue, we assume all information communication between node i and its ρ-step neighbors $j \in \mathcal{N}_i^{(\rho)}(k)$ could be finished in the period between instant k and $k+1$.*

Later, we try to propose a recursive form to calculate $y_{i,0}(k), y_{i,1}(k), \cdots, y_{i,\rho}(k)$.

3.2.2 Modelling by communication iteration

At very beginning, we could consider the local measurement $y_i(k)$ as a special case that node i only receives the data with 0-time communication. In other words, it is the case when $\xi = 0$.

$$y_{i,0}(k) = C_{i,0}x(k) + \varepsilon_{i,0}(k) \tag{3.10}$$

where $y_{i,0}(k) = y_i(k),\ C_{i,0} = C,\ \varepsilon_{i,0}(k) = v_i(k)$. In this case, the variance of noise is $\Sigma_{\varepsilon_{i,0}(k)} = \Sigma_{v_i}$.

By stacking all the sensor nodes together, we denote the global vector of the sensor data at time instant k considering 0-time communication iteration as $Y_0(k)$, which is the initial vector of later recursive computation.

$$Y_0(k) = H_0(k)X(k) + \mathcal{E}_0(k) \tag{3.11}$$

where $H_0(k) = I_{M(k)} \otimes C \in \mathcal{R}^{mM(k)\times nM(k)}$,

$$Y_0(k) = \begin{bmatrix} y_{1,0}(k) \\ y_{2,0}(k) \\ \vdots \\ y_{M(k),0}(k) \end{bmatrix} \in \mathcal{R}^{mM(k)\times 1}, X(k) = \begin{bmatrix} x(k) \\ x(k) \\ \vdots \\ x(k) \end{bmatrix} \in \mathcal{R}^{nM(k)\times 1},$$

and we get the global noise vector $\mathcal{E}_0(k) \in \mathcal{R}^{mM(k)\times 1}$ and its variance as follows

$$\mathcal{E}_0(k) = \begin{bmatrix} \varepsilon_{1,0}(k) \\ \varepsilon_{2,0}(k) \\ \vdots \\ \varepsilon_{M(k),0}(k) \end{bmatrix},\ \Sigma_{\mathcal{E}_0(k)} = \begin{bmatrix} \Sigma_{\varepsilon_{1,0}(k)} & 0 & \cdots & 0 \\ 0 & \Sigma_{\varepsilon_{2,0}(k)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{\varepsilon_{M(k),0}(k)} \end{bmatrix}$$

After 1-time communication iteration, each node i could receive the information data from its 1-step neighbors, $j \in \mathcal{N}_i^{(1)}(k)$. Then $y_{i,1}(k)$, which is the combined data in node i with 1-time communication iteration, can be proposed as follows:

$$y_{i,1}(k) = \varphi_{ii}(k)y_{i,0}(k) + \sum_{j\in\mathcal{N}_i(k)} \varphi_{ij}(k)(y_{j,0}(k) + \omega_{j,1}(k)) \tag{3.12}$$

where $\varphi_{ii}(k)$ and $\varphi_{ij}(k)$ are weighting matrices, which are designed distributively on each sensor node. $\omega_{i,\xi}(k)$ denotes the communication noise produced by node i while node i broadcasting its information to its neighbors at the ξ-th iteration between time instant k and $k+1$. We can also assume the communication noise to be a white noise, and uncorrelated with $x(k)$, $v(k)$ and $w_j(k), j \neq i$.

$$\omega_i \sim \mathcal{N}(0, \Sigma_{\omega,i}), \Sigma_{\omega,i} > 0,\ \Sigma_{\omega,i} \in \mathcal{R}^{m\times m},\ \mathcal{E}(\omega_i \omega_j^T) = \begin{cases} \Sigma_{\omega,i},\ i = j \\ 0,\ i \neq j \end{cases}, i = 1, \cdots, M(k).$$

We can also express the global sensor output vector $Y_1(k)$ after 1-time communication iteration by stacking all sensor nodes as

$$Y_1(k) = H_1(k) X(k) + \mathcal{E}_1(k) \tag{3.13}$$

where

$$H_1(k) = H_\Phi(k) H_0(k) \tag{3.14}$$

$$H_\Phi(k) = \Phi(k) (\mathcal{I}(k) + \mathcal{A}(k)) \otimes I_m \tag{3.15}$$

Here $\mathcal{A}(k) \in \mathcal{R}^{M(k)\times M(k)}$ is the adjacent matrix and $\mathcal{D}(k) \in \mathcal{R}^{M(k)\times M(k)}$ is the degree matrix of the time-varying graph. And $\Phi(k) = \{\varphi_{ij}\} \in \mathcal{R}^{M(k)\times M(k)}, i, j = 1, \cdots, M(k)$ is the weighting matrix, for each sensor node. And the local weighting matrix $\Phi_i(k)$ for distributed computation could be expressed as follows

$$\Phi_i(k) = \begin{bmatrix} \varphi_{ii}(k) & \cdots & \varphi_{ij}(k) & \cdots & \varphi_{iN_i(k)}(k) \end{bmatrix} \in \mathcal{R}^{m\times mN_i(k)} \tag{3.16}$$

And we can get the noise vector $\mathcal{E}_1(k)$ after 1-time communication iteration as

$$\mathcal{E}_1(k) = H_\Phi(k) \mathcal{E}_0(k) + \bar{H}_\Phi(k) \Omega_1(k) \tag{3.17}$$

where

$$\bar{H}_\Phi(k) = \Phi(k) \mathcal{A}(k) \otimes I_m, \tag{3.18}$$

$$Y_1(k) = \begin{bmatrix} y_{1,1}(k) \\ y_{2,1}(k) \\ \vdots \\ y_{M(k),1}(k) \end{bmatrix} \in \mathcal{R}^{mM(k)\times 1}, \mathcal{E}_1(k) = \begin{bmatrix} \varepsilon_{1,1}(k) \\ \varepsilon_{2,1}(k) \\ \vdots \\ \varepsilon_{M(k),1}(k) \end{bmatrix} \in \mathcal{R}^{mM(k)\times 1}$$

and $\Omega_1(k)$ is the vector of communication noise and its variance is expressed as.

$$\Omega_1(k) = \begin{bmatrix} \omega_{1,1}(k) \\ \omega_{2,1}(k) \\ \vdots \\ \omega_{M(k),1}(k) \end{bmatrix} \in \mathcal{R}^{mM(k)\times 1},\ \Sigma_{\Omega(k)} = \begin{bmatrix} \Sigma_{\omega,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{\omega,M(k)} \end{bmatrix} \in \mathcal{R}^{mM(k)\times mM(k)}$$

Then the variance of noise vector $\mathcal{E}_1(k)$ could be derived as

$$\Sigma_{\mathcal{E}_1(k)} = H_\Phi(k)\,\Sigma_{\mathcal{E}_0(k)} H_\Phi^T(k) + \bar{H}_\Phi(k)\,\Sigma_{\Omega(k)} \bar{H}_\Phi^T(k) \tag{3.19}$$

For each sensor node i, we can get the output data after 1-time communication iteration as

$$y_{i,1}(k) = C_{i,1}(k)x(k) + \varepsilon_{i,1}(k) \tag{3.20}$$

To achieve distributive computation on each node i, we have to find an alternative way to get $C_{i,1}(k)$ and the variance of local noise $\Sigma_{\varepsilon_{i,1}}(k)$. Sensor node i could receive $C_{j,0}(k)$, $\Sigma_{\varepsilon_{j,0}}(k)$ and $\Sigma_{\omega,j}$, $j \in \mathcal{N}_i(k)$, from its 1-step neighbors, then compose and stack all the data separately as

$$C_{i,1}(k) = \Phi_i(k) \begin{bmatrix} C_{i,0}(k) \\ \vdots \\ C_{j,0}(k) \\ \vdots \\ C_{N_i(k),0}(k) \end{bmatrix} \in \mathcal{R}^{m\times m} \tag{3.21}$$

and

$$\Sigma_{\varepsilon_{i,1}(k)} = \Phi_i(k)\Sigma_{\mathcal{E}_{i,0}(k)}\Phi_i^T(k) + \Phi_i(k)\,\Sigma_{\Omega_i(k)}\Phi_i^T(k) \tag{3.22}$$

where the variance matrices of local noise are

$$\Sigma_{\mathcal{E}_{i,0}(k)} = \begin{bmatrix} \Sigma_{\varepsilon_{i,0}(k)} & 0 & \cdots & 0 \\ 0 & \Sigma_{\varepsilon_{j,0}(k)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{\varepsilon_{Ni(k),0}(k)} \end{bmatrix}, \ \Sigma_{\Omega_i(k)} = \begin{bmatrix} \Sigma_{w_i} & 0 & \cdots & 0 \\ 0 & \Sigma_{w_j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{w_{Ni(k)}} \end{bmatrix}$$

Now we extend to ξ-time communication iteration, combining the information data by $\xi - 1$-time communication iteration from 1-step neighbors of node i, the output data of node i could be expressed as

$$y_{i,\xi}(k) = \varphi_{ii}(k) y_{i,\xi-1}(k) + \sum_{j \in \mathcal{N}_i(k)} \varphi_{ij}(k)(y_{j,\xi-1}(k) + \omega_{j,\xi}(k)) \tag{3.23}$$

In this case, the global output vector $Y_\xi(k)$ after ξ-time communication iteration is derived as follows

$$Y_\xi(k) = \begin{bmatrix} y_{1,\xi}(k) \\ y_{2,\xi}(k) \\ \vdots \\ y_{M(k),\xi}(k) \end{bmatrix} = H_\xi(k) X(k) + \mathcal{E}_\xi(k) \tag{3.24}$$

where

$$H_\xi(k) = H_\Phi^\xi(k) H_0(k) \tag{3.25}$$

$$\mathcal{E}_\xi(k) = H_\Phi^\xi(k) \mathcal{E}_0(k) + \sum_{\zeta=1}^{\xi} H_\Phi^{\xi-\zeta}(k) \bar{H}_\Phi(k) \Omega_\zeta(k) \tag{3.26}$$

Then the variance of noise vector $\mathcal{E}_1(k)$ could be derived as

$$\Sigma_{\mathcal{E}_\xi(k)} = H_\Phi^\xi(k) \Sigma_{\mathcal{E}_0(k)} (H_\Phi^\xi)^T(k) + \sum_{\zeta=1}^{\xi} H_\Phi^{\xi-\zeta}(k) \bar{H}_\Phi(k) \Sigma_{\Omega(k)} \bar{H}_\Phi^T(k) \left(H_\Phi^{\xi-\zeta}(k) \right)^T \tag{3.27}$$

The detailed derivation process is expressed in Appendix. Now we derive the distributed computation method to achieve the output data $y_{i,\xi}(k)$, observation matrix $C_{i,\xi}(k)$ and variance of the noise $\varepsilon_{i,\xi}(k)$ on each sensor node after ξ-time communication iteration.

$$y_{i,\xi}(k) = C_{i,\xi}(k) x(k) + \varepsilon_{i,\xi}(k) \tag{3.28}$$

Each sensor node i could obtain the information data that, $C_{j,\xi-1}(k)$ and $\Sigma_{\varepsilon_{j,\xi-1}}(k)$, $j \in \mathcal{N}_i(k)$, by transmitting from its 1-step neighbors. In this case, the monitoring matrix for node i by ξ-time communication iteration follows

$$C_{i,\xi}(k) = \Phi_i(k) \begin{bmatrix} C_{i,\xi-1}(k) \\ \vdots \\ C_{j,\xi-1}(k) \\ \vdots \\ C_{N_i(k),\xi-1}(k) \end{bmatrix} \in \mathcal{R}^{m\times m} \tag{3.29}$$

and

$$\Sigma_{\varepsilon_{i,\xi}(k)} = \Phi_i(k)\Sigma_{\mathcal{E}_{i,\xi-1}(k)}\Phi_i^T(k) + \Phi_i(k)\,\Sigma_{\Omega_i(k)}\Phi_i^T(k) \tag{3.30}$$

where the variance matrices of local noise are

$$\Sigma_{\mathcal{E}_{i,\xi-1}(k)} = \begin{bmatrix} \Sigma_{\varepsilon_{i,\xi-1}(k)} & 0 & \cdots & 0 \\ 0 & \Sigma_{\varepsilon_{j,\xi-1}(k)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{\varepsilon_{Ni(k),\xi-1}(k)} \end{bmatrix} \in \mathcal{R}^{m(N_i(k)+1)\times m(N_i(k)+1)}$$

The communication iteration for each node will be continued, until $\xi = \rho$. The problem how to determine the maximum iteration time ρ will be discussed later depending on the variance of estimation error for each sensor agent.

We summarize the algorithm of communication modelling in Algorithm.2 in a distributed and recursive form.

3.3 Distributed H_2 Observer Design in a Recursive Form

We have to design a distributed H_2 observer at node i to estimate $x(k+1)$, $d(k)$ by using the data from local measurement $y_i(0), y_i(1), \cdots, y_i(k)$, and also by communication with its neighbors $y_{i,\xi}(k)$, $\xi = 0, 1, \cdots, \rho$. As we known, more data could lead to a better estimation performance.

Now, We would like to find the solution by solving an optimization problem.

Algorithm 2 Communication modelling in a recursive form

Step 1: Local measurement and initial data transmission

1: **for** $i = 1, \cdots, M(k)$ **do**
2: each sensor achieves the local measurement $y_{i,0}(k)$;
3: each sensor transmits its information (local measurement $y_{i,0}(k)$, variance of measurement noise Σ_{v_i}, variance of communication noise Σ_{w_i} and monitoring matrix $C_{i,0}(k)$) to its 1-step neighbors;
4: Set $\xi = 1$
5: **end for**

Step 2: Receive information

1: **for** $i = 1, \cdots, M(k)$ **do**
2: Each nodes receive information from its neighbors
3: Count the number of neighbors $N_i^{(1)}$ and collect topology data
4: Compute weighting matrices $\varphi_{ii}(k)$, $\varphi_{ij}(k)$ and $\Phi_i(k)$
5: **end for**

Step 3: Data computation and transmission

1: **for** $i = 1, \cdots, M(k)$ **do**
2: Calculate $y_{i,\xi}(k)$ under the communication law

$$y_{i,\xi}(k) = \varphi_{ii}(k) y_{i,\xi-1}(k) + \sum_{j \in \mathcal{N}_i(k)} \varphi_{ij}(k)(y_{j,\xi-1}(k) + \omega_{j,\xi}(k))$$

3: Compose $\Sigma_{\mathcal{E}_{i,\xi-1}(k)}$ and $\Sigma_{\Omega_i(k)}$;
4: Calculate variance matrix under the communication law

$$\Sigma_{\varepsilon_{i,\xi}(k)} = \Phi_i(k) \Sigma_{\mathcal{E}_{i,\xi-1}(k)} \Phi_i^T(k) + \Phi_i(k) \Sigma_{\Omega_i(k)} \Phi_i^T(k)$$

5: Calculate new monitoring matrix $C_{i,\xi}(k)$ by Eq.3.29
6: Information ($y_{i,\xi}(k)$, $\Sigma_{\varepsilon_{i,\xi}(k)}$, Σ_{w_i} and monitoring matrix $C_{i,\xi}(k)$) transmission;
7: **end for**

Step 4: Check and repeat

1: if $\xi = \rho$, the communication iteration will stop;
2: else set $\xi = \xi + 1$ Repeat from **Step 2** to **Step 4**

The cost function J_i at node i is

$$\min_{d(k),x(k)} \begin{pmatrix} \frac{1}{2}x^T(0)P_{i,0}^{-1}(0)x(0) + \frac{1}{2}\sum_{k=0}^{N-1} d^T(k)d(k) + \\ \frac{1}{2}\sum_{k=0}^{N}\sum_{\xi=0}^{\rho}\Big((y_{i,\xi}(k) - C_{i,\xi}(k)x(k))^T\Sigma^{-1}_{\varepsilon_{i,\xi}(k)}(y_{i,\xi}(k) - C_{i,\xi}(k)x(k))\Big) \end{pmatrix} \tag{3.31}$$

where $P_{i,0}(0)$ is the initial value of the variance of the estimation error at node i. After ξ-time communication iteration, it is denoted by $P_{i,\xi}(k)$. Another expression of the cost function of node i follows as

$$J_i = \min_{d(k),x(k)} \begin{pmatrix} \frac{1}{2}\left\|x(0)\right\|^2_{P_{i,0}^{-1}(0)} + \frac{1}{2}\left\|d(k)\right\|^2_{2,[0,N-1]} \\ +\frac{1}{2}\sum_{\xi=0}^{\rho}\left\|y_{i,\xi}(k) - C_{i,\xi}(k)x(k)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,\xi}(k)},[0,N]} \end{pmatrix} \tag{3.32}$$

Later, we will use induction method to solve the distributed estimation problem.

3.3.1 Starting at time period $k = [0, 1)$

Without communication iteration

At very beginning, the $i-th$ node could only get the local measurement $y_{i,0}(0)$. The cost function is

$$J_i(x(0)) = \frac{1}{2}\left\|x(0) - \hat{x}_{i,0}(0)\right\|^2_{P_{i,0}^{-1}(0)} + \frac{1}{2}\left\|y_{i,0}(0) - C_{i,0}(0)x(0)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(0)}} \tag{3.33}$$

with the initial value $\hat{x}_{i,0}(0) = 0$, $P_{i,0}(0) = I$. We would like to find $\hat{x}_{i,0}(0|0)$ for given $y_{i,0}(0)$.

$$J^0_{i,0}(\hat{x}_{i,0}(0|0)) = \min_{x(0)} J^0_{i,0}(x(0)) \tag{3.34}$$

To solve the optimization problem, we derive that

$$\frac{\partial J^0_{i,0}(x(0))}{\partial x(0)} = P^{-1}_{i,0}(0)x(0) - C^T_{i,0}(0)\Sigma^{-1}_{\varepsilon_{i,0}(0)}(y_{i,0}(0) - C_{i,0}(0)x(0)) = 0 \Rightarrow$$

Then

$$\hat{x}_{i,0}(0|0) = (P^{-1}_{i,0}(0) + C^T_{i,0}(0)\Sigma^{-1}_{\varepsilon_{i,0}(0)}C_{i,0}(0))^{-1}C^T_{i,0}(0)\Sigma^{-1}_{\varepsilon_{i,0}(0)}y_{i,0}(0) \tag{3.35}$$

Let

$$P_{i,0}(0|0) = (P^{-1}_{i,0}(0) + C^T_{i,0}(0)\Sigma^{-1}_{\varepsilon_{i,0}(0)}C_{i,0}(0))^{-1} \tag{3.36}$$

For ease of numerical computation, the expression for Eq.3.36 may be written as:

$$P_{i,0}(0|0) = P_{i,0}(0) - P_{i,0}(0)C_{i,0}^T(0)(\Sigma_{\varepsilon_{i,0}(0)} + C_{i,0}(0)P_{i,0}(0)C_{i,0}^T(0))^{-1}C_{i,0}(0)P_{i,0}(0) \quad (3.37)$$

We could derive the state estimation on node i, Eq.3.35 as follows

$$\hat{x}_{i,0}(0|0) = \hat{x}_{i,0}(0) + K_{i,0}(0)(y_{i,0}(0) - C_{i,0}\hat{x}_{i,0}(0)) \quad (3.38)$$

where $K_{i,0}(0)$ is the observer gain matrix.

$$K_{i,0}(0) = P_{i,0}(0)C_{i,0}^T(0)(\Sigma_{\varepsilon_{i,0}(0)} + C_{i,0}(0)P_{i,0}(0)C_{i,0}^T(0))^{-1} \quad (3.39)$$

and the updated variance of estimation error is expressed as

$$P_{i,0}(0|0) = P_{i,0}(0) - K_{i,0}(0)\Psi_{i,0}(0|0)K_{i,0}^T(0) \quad (3.40)$$

Here, we denote $r_{i,0}(0)$ as residual signal generated in node i. Moreover, the residual signal could be used for distributed fault detection, which will be discussed later. It follows

$$r_{i,0}(0) = y_{i,0}(0) - C_{i,0}\hat{x}_{i,0}(0) \quad (3.41)$$

where $\Psi_{i,0}(0|0)$ denotes the variance of residual signal. It is a routine computation to show

$$\Psi_{i,0}(0|0) = \Sigma_{\varepsilon_{i,0}(0)} + C_{i,0}(0)P_{i,0}(0)C_{i,0}^T(0) \quad (3.42)$$

Due to an alternative interpretation of the cost function, then we can derive Eq.3.33 as

$$J_{i,0}^0(x(0)) = J_{i,const}^0(0) + \frac{1}{2}\|x(0) - \hat{x}_{i,0}(0|0)\|^2_{P_{i,0}^{-1}(0|0)} \quad (3.43)$$

where

$$J_{i,const}^0(0) = \frac{1}{2}\|\hat{x}_{i,0}(0|0)\|^2_{P_{i,0}^{-1}(0)} + \frac{1}{2}\|y_{i,0}(0) - C_{i,0}(0)\hat{x}_{i,0}(0|0)\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(0)}} \quad (3.44)$$

The detailed derivation from Eq.3.33 to Eq.3.43 is expressed in Appendix.

When communication iteration ξ equals 1

Next, we consider the situation that $k = 0$ and $\xi = 1$. In this case, the sensor node i can not only obtain the local sensor data $y_{i,0}$, but also receive the information data from all its 1-step neighbors. According to Eq.3.32 and Eq.3.43, the cost function of node i could

be expressed as follows

$$J^1_{i,0}(x(0)) = \begin{pmatrix} J^0_{i,const}(0) + \frac{1}{2}\left\|x(0) - \hat{x}_{i,0}(0|0)\right\|^2_{P^{-1}_{i,0}(0|0)} \\ +\frac{1}{2}\left\|y_{i,1}(0) - C_{i,1}(0)x(0)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,1}(0)}} \end{pmatrix} \tag{3.45}$$

We would like to find $\hat{x}_{i,1}(0|0)$ for given $y_{i,0}(0),\ y_{i,1}(0)$.

$$J^1_{i,0}(\hat{x}_{i,1}(0|0)) = \min_{x(0)} J^1_{i,0}(x(0))$$

to solve the optimization problem, that

$$\frac{\partial J^1_{i,0}(x(0))}{\partial x(0)} = P^{-1}_{i,0}(0|0)(x(0) - \hat{x}_{i,0}(0|0)) - C^T_{i,1}(0)\Sigma^{-1}_{\varepsilon_{i,1}(0)}(y_{i,1}(0) - C_{i,1}(0)x(0)) = 0 \Rightarrow$$

Then

$$\hat{x}_{i,1}(0|0) = P_{i,1}(0|0)(C^T_{i,1}(0)\Sigma^{-1}_{\varepsilon_{i,1}(0)}y_{i,1}(0) + P^{-1}_{i,0}(0|0)\hat{x}_{i,0}(0|0)) \tag{3.46}$$

where

$$\begin{aligned} P_{i,1}(0|0) &= (P^{-1}_{i,0}(0|0) + C^T_{i,1}(0)\Sigma^{-1}_{\varepsilon_{i,1}(0)}C_{i,1}(0))^{-1} \\ &= P_{i,0}(0|0) - P_{i,0}(0|0)C^T_{i,1}(0)(\Sigma_{\varepsilon_{i,1}(0)} + C_{i,1}(0)P_{i,0}(0|0)C^T_{i,1}(0))^{-1}C_{i,1}(0)P_{i,0}(0|0) \end{aligned}$$

We can get

$$\hat{x}_{i,1}(0|0) = \hat{x}_{i,0}(0|0) + K_{i,1}(0)(y_{i,1}(0) - C_{i,1}(0)\hat{x}_{i,0}(0|0)) \tag{3.47}$$

where the updated observer gain matrix, variance of estimation error and variance of residual signal could be derived as follows

$$K_{i,1}(0) = P_{i,0}(0|0)C^T_{i,1}(0)(\Sigma_{\varepsilon_{i,1}(0)} + C_{i,1}(0)P_{i,0}(0|0)C^T_{i,1}(0))^{-1} \tag{3.48}$$

$$P_{i,1}(0|0) = P_{i,0}(0|0) - K_{i,1}(0)\Psi_{i,1}(0|0)K^T_{i,1}(0) \tag{3.49}$$

$$\Psi_{i,1}(0|0) = \Sigma_{\varepsilon_{i,1}(0)} + C_{i,1}(0)P_{i,0}(0|0)C^T_{i,1}(0) \tag{3.50}$$

We can get the residual signal after 1-time communication iteration as

$$r_{i,1}(0) = y_{i,1}(0) - C_{i,1}\hat{x}_{i,0}(0) \tag{3.51}$$

Then, we can derive the updated cost function of node i after 1-time communication iteration as

$$J_{i,0}^{1}(x(0)) = J_{i,const}^{1}(0) + \frac{1}{2}\left\|x(0) - \hat{x}_{i,1}(0|0)\right\|_{P_{i,1}^{-1}(0|0)}^{2} \tag{3.52}$$

where $J_{i,const}^{1}$ is the term unrelated with $x(0)$.

When communication iteration ξ equals ρ

Now, we assume the case in k = 0 ,$\xi = \rho - 1$ is true. We can extend to the situation that $k = 0$ and $\xi = \rho$. According to Eq.3.32, the cost function could be formulated as

$$J_{i,0}^{\rho}(x(0)) = \frac{1}{2}\left\|x(0) - \hat{x}_{i,0}(0)\right\|_{P_{i,0}^{-1}(0)}^{2} + \frac{1}{2}\sum_{\xi=0}^{\rho}\left\|y_{i,\xi}(0) - C_{i,\xi}(0)x(0)\right\|_{\Sigma_{\varepsilon_{i,\xi}}^{-1}(0)}^{2} \tag{3.53}$$

Based on the cost function of node i by $\rho - 1$-time communication iteration, then the Eq.3.53 could be derived as

$$J_{i,0}^{\rho}(x(0)) = J_{i,const}^{\rho-1}(0) + \frac{1}{2}\left\|x(0) - \hat{x}_{i,\rho-1}(0|0)\right\|_{P_{i,\rho-1}^{-1}(0|0)}^{2} + \frac{1}{2}\left\|y_{i,\rho}(0) - C_{i,\rho}(0)x(0)\right\|_{\Sigma_{\varepsilon_{i,\rho}}^{-1}(0)}^{2}$$

In order to find $\hat{x}_{i,\rho}(0|0)$ for given $y_{i,0}(0),\ y_{i,1}(0), \cdots, y_{i,\rho}(0)$.

$$J_{i,0}^{\rho}(\hat{x}_{i,\rho}(0|0)) = \min_{x(0)} J_{i,0}^{\rho}(x(0))$$

to solve the optimization problem below that

$$\begin{aligned}\frac{\partial J_{i,0}^{\rho}(x(0))}{\partial x(0)} &= P_{i,\rho-1}^{-1}(0|0)(x(0) - \hat{x}_{i,\rho-1}(0|0)) - C_{i,\rho}^{T}(0)\Sigma_{\varepsilon_{i,\rho}(0)}^{-1}(y_{i,\rho}(0) - C_{i,\rho}(0)x(0)) \\ &= 0 \Rightarrow\end{aligned}$$

Then the state estimation follows

$$\hat{x}_{i,\rho}(0|0) = P_{i,\rho}(0|0)(C_{i,\rho}^{T}(0)\Sigma_{\varepsilon_{i,\rho}(0)}^{-1}y_{i,\rho}(0) + P_{i,\rho-1}^{-1}(0|0)\hat{x}_{i,\rho-1}(0|0)) \tag{3.54}$$

where

$$P_{i,\rho}(0|0) = P_{i,\rho-1}^{-1}(0|0) - K_{i,\rho}(0)\Psi_{i,\rho}(0|0)K_{i,\rho}^{T}(0) \tag{3.55}$$

$$\Psi_{i,\rho}(0|0) = \Sigma_{\varepsilon_{i,\rho}(0)} + C_{i,\rho}(0)P_{i,\rho-1}^{-1}(0|0)C_{i,\rho}^{T}(0) \tag{3.56}$$

$$K_{i,\rho}(0) = P_{i,\rho-1}^{-1}(0|0)C_{i,\rho}^{T}(0)(\Sigma_{\varepsilon_{i,\rho}(0)} + C_{i,\rho}(0)P_{i,\rho-1}^{-1}(0|0)C_{i,\rho}^{T}(0))^{-1} \tag{3.57}$$

The state estimation by Eq.3.54 is written as

$$\hat{x}_{i,\rho}(0|0) = \hat{x}_{i,\rho-1}(0|0) + K_{i,\rho}(0)(y_{i,\rho}(0) - C_{i,\rho}(0)\hat{x}_{i,\rho-1}(0|0)) \tag{3.58}$$

We can get the residual signal after ρ-time communication iteration as

$$r_{i,\rho}(0) = y_{i,\rho}(0) - C_{i,\rho}\hat{x}_{i,\rho-1}(0|0) \tag{3.59}$$

Then the cost function of node i at time instant $k = 0$ after ρ-time iteration can be derived as follows

$$J^{\rho}_{i,0}(x(0)) = J^{\rho}_{i,const}(0) + \frac{1}{2}\left\|x(0) - \hat{x}_{i,\rho}(0|0)\right\|^2_{P^{-1}_{i,\rho}(0|0)} \tag{3.60}$$

As the result, we summarize the algorithm of H_2 observer design when $k = 0$ in Algorithm.3.

3.3.2 During time period $k = [1, 2)$

Without communication iteration

Now, we extend to the situation when $k = 1$ without communication iteration. According to Eq.3.32, the cost function of node i follows

$$\begin{aligned} J^0_{i,1}(x(1), d(0)) = &\frac{1}{2}\left\|x(0) - \hat{x}_{i,0}(0)\right\|^2_{P^{-1}_{i,0}(0)} + \frac{1}{2}\sum_{\xi=0}^{\rho}\left\|y_{i,\xi}(0) - C_{i,\xi}(0)x(0)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,\xi}(0)}} \\ &+ \frac{1}{2}\left\|y_{i,0}(1) - C_{i,0}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(1)}} + \frac{1}{2}\left\|d(0)\right\|^2 \end{aligned}$$

Then the cost function of node i could be derived as

$$J^0_{i,1}(x(1), d(0)) = \begin{pmatrix} J^{\rho}_{i,const}(0) + \frac{1}{2}\left\|x(0) - \hat{x}_{i,\rho}(0|0)\right\|^2_{P^{-1}_{i,\rho}(0|0)} \\ + \frac{1}{2}\left\|y_{i,0}(1) - C_{i,0}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(1)}} + \frac{1}{2}\left\|d(0)\right\|^2 \end{pmatrix} \tag{3.61}$$

We would like to find $\hat{x}_{i,0}(1|1)$ and $\hat{d}_{i,0}(0)$ for given $y_{i,0}(0),\ y_{i,1}(0), \cdots, y_{i,\rho}(0)$ and $y_{i,0}(1)$.

$$J^0_{i,1}(\hat{x}_{i,1}(0|0), \hat{d}_{i,0}(0)) = \min_{x(1),d(0)} J^0_{i,1}(x(1), d(0))$$

to solve the optimization problem.

Due to $x(1) = Ax(0) + d(0)$, we can get $x(0) = A^{-1}x(1) - A^{-1}d(0)$. Then we

Algorithm 3 Distributed H_2 observer in a recursive form at $k = 0$

Step 1: Local measurement and estimation

for $i = 1, \cdots, M(0)$ **do**
  Each sensor obtains the local measurement $y_{i,0}(0)$;
  Set initial value that $\hat{x}_{i,0}(0) = 0$, $P_{i,0}(0) = I$ and $\xi = 0$
  Computation: $K_{i,0}(0) = P_{i,0}(0)C_{i,0}^T(0)(\Sigma_{\varepsilon_{i,0}(0)} + C_{i,0}(0)P_{i,0}(0)C_{i,0}^T(0))^{-1}$;
  Estimation: $\hat{x}_{i,0}(0|0) = \hat{x}_{i,0}(0) + K_{i,0}(0)(y_{i,0}(0) - C_{i,0}\hat{x}_{i,0}(0))$;
  Residual signal generation: $r_{i,0}(0) = y_{i,0}(0) - C_{i,0}\hat{x}_{i,0}(0)$
  Computation: Variance of residual signal $\Psi_{i,0}(0|0) = \Sigma_{\varepsilon_{i,0}(0)} + C_{i,0}(0)P_{i,0}(0)C_{i,0}^T(0)$
  Computation: Variance of estimation error
  $P_{i,0}(0|0) = P_{i,0}(0) - K_{i,0}(0)\Psi_{i,0}(0|0)K_{i,0}^T(0)$
end for

Step 2: Information transmission

for $i = 1, \cdots, M(0)$ **do**
  each sensor transmits its information ($y_{i,\xi}(0)$, variance of noise $\Sigma_{\varepsilon_{i,\xi}(0)}$, Σ_{w_i} and monitoring matrix $C_{i,\xi}(0)$) to its 1-step neighbors;
end for

Step 3: Receive information

for $i = 1, \cdots, M(0)$ **do**
  Each nodes receive information from its neighbors
  Count the number of neighbors $N_i^{(1)}(0)$
  Compute weighting matrices $\varphi_{ii}(k)$, $\varphi_{ij}(k)$ and $\Phi_i(k)$
end for

Step 4: Update information

for $i = 1, \cdots, M(0)$ **do**
  Calculation $y_{i,\xi}(k) = \varphi_{ii}(k)y_{i,\xi-1}(k) + \sum_{j\in\mathcal{N}_i(k)}\varphi_{ij}(k)(y_{j,\xi-1}(k) + \omega_{j,\xi}(k))$
  Compose $\Sigma_{\mathcal{E}_{i,\xi-1}(k)}$ and $\Sigma_{\Omega_i(k)}$;
  Calculate variance matrix under the communication law
  $\Sigma_{\varepsilon_{i,\xi}(k)} = \Phi_i(k)\Sigma_{\mathcal{E}_{i,\xi-1}(k)}\Phi_i^T(k) + \Phi_i(k)\Sigma_{\Omega_i(k)}\Phi_i^T(k)$
  Calculate new monitoring matrix $C_{i,\xi}(k)$ by Eq.3.29
end for

Step 5: Update estimation

for $i = 1, \cdots, M(0)$ **do**
  Computation : $K_{i,\xi}(0) = P_{i,\xi-1}(0|0)C_{i,\xi}^T(0)(\Sigma_{\varepsilon_{i,\xi}(0)} + C_{i,\xi}(0)P_{i,\xi-1}(0|0)C_{i,\xi}^T(0))^{-1}$;
  Estimation: $\hat{x}_{i,\xi}(0) = \hat{x}_{i,\xi-1}(0) + K_{i,\xi}(0)(y_{i,\xi}(0) - C_{i,\xi}(0)\hat{x}_{i,\xi-1}(0))$;
  Residual signal generation: $r_{i,\xi}(0) = y_{i,\xi}(0) - C_{i,\xi}\hat{x}_{i,\xi-1}(0)$;
  Computation : Variance of residual signal
  $\Psi_{i,\xi}(0|0) = \Sigma_{\varepsilon_{i,\xi}(0)} + C_{i,\xi}(0)P_{i,\xi-1}(0)C_{i,\xi}^T(0)$
  Computation : Variance of estimation error
  $P_{i,\xi}(0|0) = P_{i,\xi-1}(0) - K_{i,\xi}(0)\Psi_{i,\xi}(0|0)K_{i,\xi}^T(0)$
end for

Step 6: Repeat computation

Repeat from **Step 2** to **Step 5** until $\xi = \rho$
Set $\xi = \xi + 1$

substitute it to Eq.3.61, so that

$$J_{i,1}^{0}(x(1),d(0)) = \begin{pmatrix} J_{i,const} + \frac{1}{2}\left\|A^{-1}x(1) - A^{-1}d(0) - \hat{x}_{i,\rho}(0|0)\right\|^2_{P_{i,\rho}^{-1}(0|0)} \\ +\frac{1}{2}\left\|y_{i,0}(1) - C_{i,0}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(1)}} + \frac{1}{2}\left\|d(0)\right\|^2 \end{pmatrix} \tag{3.62}$$

Due to optimization issue, it follows

$$\begin{aligned} \frac{\partial J_{i,1}^{0}(x(1),d(0))}{\partial d(0)} &= d(0) - A^{-T}P_{i,\rho}^{-1}(0|0)(A^{-1}x(1) - A^{-1}d(0) - \hat{x}_{i,\rho}(0|0)) \\ &= 0 \Rightarrow, \end{aligned}$$

we can achieve

$$\hat{d}_{i,0}(0) = P_{i,0}^{-1}(1|0)(x(1) - A\hat{x}_{i,\rho}(0|0)) \tag{3.63}$$

where

$$P_{i,0}(1|0) = I + AP_{i,\rho}(0|0)A^T \tag{3.64}$$

And we can get

$$\left\|x(1) - A\hat{x}_{i,\rho}(0|0)\right\|^2_{P_{i,0}^{-1}(1|0)} = \left\|A^{-1}x(1) - A^{-1}d(0) - \hat{x}_{i,\rho}(0|0)\right\|^2_{P_{i,\rho}^{-1}(0|0)} + \left\|d(0)\right\|^2 \tag{3.65}$$

We may find the optimal solution for $x(1)$ that

$$J_{i,1}^{0}(x(1),d(0)) = J_{i,const}^{\rho}(0) + \frac{1}{2}\left\|x(1) - A\hat{x}_{i,\rho}(0|0)\right\|^2_{P_{i,0}^{-1}(1|0)} + \frac{1}{2}\left\|y_{i,0}(1) - C_{i,0}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(1)}}$$

To achieve optimization, the partial derivative of $J_{i,1}^{0}(x(1),d(0))$ with respect to the state vector $x(1)$ is denoted by

$$\frac{\partial J_{i,1}^{0}(x(1),d(0))}{\partial x(1)} = P_{i,0}^{-1}(1|0)(x(1) - A\hat{x}_{i,\rho}(0|0)) - C_{i,0}^{T}(1)\Sigma^{-1}_{\varepsilon_{i,0}(1)}(y_{i,0}(1) - C_{i,0}(1)x(1)).$$

It leads to

$$\hat{x}_{i,0}(1|1) = P_{i,0}(1|1)(C_{i,1}^{T}(0)\Sigma^{-1}_{\varepsilon_{i,0}(1)}y_{i,0}(1) - P_{i,0}^{-1}(1|0)\hat{x}_{i,0}(1|0)) \tag{3.66}$$

where

$$P_{i,0}(1|1) = (P_{i,0}^{-1}(1|0) + C_{i,0}^T(1)\Sigma_{\varepsilon_{i,0}(1)}^{-1}C_{i,0}(1))^{-1} \tag{3.67}$$

$$\hat{x}_{i,0}(1|0) = A\hat{x}_{i,\rho}(0|0) \tag{3.68}$$

The updated variance of estimation error and estimated state could be written in a recursive form by

$$P_{i,0}(1|1) = P_{i,0}(1|0) - K_{i,0}(1)\Psi_{i,0}(1|1)K_{i,0}^T(1) \tag{3.69}$$

$$\Psi_{i,0}(1|1) = \Sigma_{\varepsilon_{i,0}(1)} + C_{i,0}(1)P_{i,0}(1|0)C_{i,0}^T(1) \tag{3.70}$$

$$\hat{x}_{i,0}(1|1) = \hat{x}_{i,0}(1|0) + K_{i,0}(1)(y_{i,0}(1) - C_{i,0}(1)\hat{x}_{i,0}(1|0)) \tag{3.71}$$

where

$$K_{i,0}(1) = P_{i,0}(1|0)C_{i,0}^T(1)(\Sigma_{\varepsilon_{i,0}(1)} + C_{i,0}(1)P_{i,0}(1|0)C_{i,0}^T(1))^{-1} \tag{3.72}$$

The residual signal can be denoted as follows

$$r_{i,0}(1) = y_{i,0}(1) - C_{i,0}(1)\hat{x}_{i,0}(1|0) \tag{3.73}$$

And we can substitute Eq.3.71 to Eq.3.63, $\hat{d}_{i,0}(0)$ is derived as

$$\begin{aligned}\hat{d}_{i,0}(0) &= P_{i,0}^{-1}(1|0)(\hat{x}_{i,0}(1|1) - \hat{x}_{i,0}(1|0)) \\ &= P_{i,0}^{-1}(1|0)K_{i,0}(1)(y_{i,0}(1) - C_{i,0}(1)\hat{x}_{i,0}(1|0)) \\ &= C_{i,0}^T(1)(\Sigma_{\varepsilon_{i,0}(1)} + C_{i,0}(1)P_{i,0}(1|0)C_{i,0}^T(1))^{-1}(y_{i,0}(1) - C_{i,0}(1)\hat{x}_{i,0}(1|0))\end{aligned}$$

We can get

$$\hat{d}_{i,0}(0) = C_{i,0}^T(1)\Psi_{i,0}^{-1}(1|1)(y_{i,0}(1) - C_{i,0}(1)\hat{x}_{i,0}(1|0)) \tag{3.74}$$

Now, we could write the cost function of node i as follows

$$J_{i,1}^0(x(1), d(0)) = J_{i,const}^0(1) + \frac{1}{2}\left\|x(1) - \hat{x}_{i,0}(1|1)\right\|_{P_{i,0}^{-1}(1|1)}^2 \tag{3.75}$$

where

$$J_{i,const}^0(1) = J_{i,const}^\rho(0) + \frac{1}{2}\left\|\hat{x}_{i,0}(1|1) - \hat{x}_{i,0}(1|0)\right\|_{P_{i,0}^{-1}(1|0)}^2 + \frac{1}{2}\left\|y_{i,0}(1) - C_{i,0}(1)\hat{x}_{i,0}(1|1)\right\|_{\Sigma_{\varepsilon_{i,0}(1)}^{-1}}^2$$

Later, we consider the situation with communication iteration.

When communication iteration ξ equals 1

Here, it is the situation that $k = 1$ and $\xi = 1$, which means only 1-step communication iteration is considered. The cost function is

$$J^1_{i,1}(x(1), d(0)) = \begin{pmatrix} J^0_{i,const}(1) + \frac{1}{2}\left\|x(1) - \hat{x}_{i,0}(1|1)\right\|^2_{P^{-1}_{i,0}(1|1)} \\ +\frac{1}{2}\left\|y_{i,1}(1) - C_{i,1}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,1}(1)}} \end{pmatrix} \tag{3.76}$$

We would like to find $\hat{x}_{i,1}(1|1)$ and $\hat{d}_{i,1}(0)$ for given $y_{i,0}(0),\ y_{i,1}(0), \cdots, y_{i,\rho}(0)$ and $y_{i,0}(1)$, $y_{i,1}(1)$ to achieve

$$J^1_{i,1}(\hat{x}_{i,1}(1|1), \hat{d}_{i,1}(0)) = \min_{x(1), d(0)} J^1_{i,1}(x(1), d(0))$$

In order to solve the optimization problem, the partial derivative of the cost function with respect to the state $x(1)$ is denoted as

$$\begin{aligned} \frac{\partial J^1_{i,1}(x(1), d(0))}{\partial x(1)} &= P^{-1}_{i,0}(1|1)(x(1) - \hat{x}_{i,0}(1|1)) - C^T_{i,1}(1)\Sigma^{-1}_{\varepsilon_{i,1}(1)}(y_{i,1}(1) - C_{i,1}(1)x(1)) \\ &= 0 \Rightarrow \end{aligned}$$

Then we could obtain

$$\hat{x}_{i,1}(1|1) = \hat{x}_{i,0}(1|1)) + K_{i,1}(1)(y_{i,1}(1) - C_{i,1}(1)\hat{x}_{i,0}(1|1)) \tag{3.77}$$

where

$$K_{i,1}(1) = P_{i,0}(1|1)C^T_{i,1}(1)(\Sigma_{\varepsilon_{i,1}(1)} + C_{i,1}(1)P_{i,0}(1|1)C^T_{i,1}(1))^{-1} \tag{3.78}$$

$$P_{i,1}(1|1) = P_{i,0}(1|1) - K_{i,1}(1)\Psi_{i,1}(1|1)K^T_{i,1}(1) \tag{3.79}$$

$$\Psi_{i,1}(1|1) = \Sigma_{\varepsilon_{i,1}(1)} + C_{i,1}(1)P_{i,0}(1|1)C^T_{i,1}(1) \tag{3.80}$$

The residual signal could be expressed as follows

$$r_{i,1}(1) = y_{i,1}(1) - C_{i,1}(1)\hat{x}_{i,0}(1|1) \tag{3.81}$$

And the estimation of disturbance $\hat{d}_{i,1}(0)$ could be derived by

$$\hat{d}_{i,1}(0) = \hat{d}_{i,0}(0) + C^T_{i,1}(1)\Psi^{-1}_{i,1}(1|1)(y_{i,1}(1) - C_{i,1}(1)\hat{x}_{i,0}(1|1))) \tag{3.82}$$

Due to Eq.3.76, then we get the cost function as follows

$$\begin{aligned} J^1_{i,1}(x(1),d(0)) &= J^0_{i,const}(1) + \frac{1}{2}\left\|x(1)-\hat{x}_{i,0}(1|1)\right\|^2_{P^{-1}_{i,0}(1|1)} + \frac{1}{2}\left\|y_{i,1}(1)-C_{i,1}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,1}(1)}} \\ &= J^1_{i,const}(1) + \frac{1}{2}\left\|x(1)-\hat{x}_{i,1}(1|1)\right\|^2_{P^{-1}_{i,1}(1|1)} \end{aligned}$$

When communication iteration ξ equals ρ

We assume the case in k = 1, $\xi = \rho - 1$ is true. Next, we can extend the situation to $k = 1, \xi = \rho$. According to Eq.3.32, the cost function is derived as

$$\begin{aligned} J^\rho_{i,1}(x(1),d(0)) = \frac{1}{2}\left\|x(0)-\hat{x}_{i,0}(0)\right\|^2_{P^{-1}_{i,0}(0)} + \frac{1}{2}\sum_{\xi=0}^{\rho}\left\|y_{i,\xi}(0)-C_{i,\xi}(0)x(0)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,\xi}(0)}} \\ + \frac{1}{2}\sum_{\xi=0}^{\rho}\left\|y_{i,\xi}(1)-C_{i,\xi}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,\xi}(1)}} + \frac{1}{2}\left\|d(0)\right\|^2 \end{aligned}$$

We can get

$$J^\rho_{i,1}(x(1),d(0)) = \begin{pmatrix} J^{\rho-1}_{i,const}(1) + \frac{1}{2}\left\|x(1)-\hat{x}_{i,\rho-1}(1|1)\right\|^2_{P^{-1}_{i,\rho-1}(1|1)} \\ +\frac{1}{2}\left\|y_{i,\rho}(1)-C_{i,\rho}(1)x(1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,\rho}(1)}} \end{pmatrix} \tag{3.83}$$

As the same with the situation we mentioned before, we have to find $\hat{x}_{i,\rho}(1|1)$ and $\hat{d}_{i,\rho}(0)$ by using $y_{i,0}(0),\ y_{i,1}(0),\cdots,y_{i,\rho}(0)$ and also $y_{i,0}(1),\ y_{i,1}(1),\cdots,y_{i,\rho}(1)$.

$$J^\rho_{i,1}(\hat{x}_{i,\rho}(1|1),\hat{d}_{i,\rho}(0)) = \min_{x(1),d(0)} J^\rho_{i,1}(x(1),d(0))$$

to solve the optimization problem, which leads to

$$\begin{aligned} \frac{\partial J^\rho_{i,1}(x(1),d(0))}{\partial x(1)} &= P^{-1}_{i,\rho-1}(1|1)\left(x(1)-\hat{x}_{i,\rho-1}(1|1)\right) - C^T_{i,\rho}(1)\Sigma^{-1}_{\varepsilon_{i,\rho}(1)}(y_{i,\rho}(1)-C_{i,\rho}(1)x(1)) \\ &= 0 \Rightarrow \end{aligned}$$

It turns out

$$\hat{x}_{i,\rho}(1|1) = \hat{x}_{i,\rho-1}(1|1) + K_{i,\rho}(1)(y_{i,\rho}(1) - C_{i,\rho}(1)\hat{x}_{i,\rho-1}(1|1)) \tag{3.84}$$

where

$$K_{i,\rho}(1) = P_{i,\rho-1}(1|1)C_{i,\rho}^T(1)(\Sigma_{\varepsilon_{i,\rho}(1)} + C_{i,\rho}(1)P_{i,\rho-1}(1|1)C_{i,\rho}^T(1))^{-1} \tag{3.85}$$

$$\Psi_{i,\rho}(1|1) = \Sigma_{\varepsilon_{i,\rho}(1)} + C_{i,\rho}(1)P_{i,\rho-1}(1|1)C_{i,\rho}^T(1) \tag{3.86}$$

$$P_{i,\rho}(1|1) = P_{i,\rho-1}(1|1) - K_{i,\rho}(1)\Psi_{i,\rho}(1|1)K_{i,\rho}^T(1) \tag{3.87}$$

The residual signal is given by

$$r_{i,\rho}(1) = y_{i,\rho}(1) - C_{i,\rho}(1)\hat{x}_{i,\rho-1}(1|1) \tag{3.88}$$

Then $\hat{d}_{i,0}(\rho)$, the estimation of disturbance is given by

$$\begin{aligned}\hat{d}_{i,\rho}(0) &= P_{i,0}^{-1}(1|0)(\hat{x}_{i,\rho}(1|1) - A\hat{x}_{i,\rho}(0|0)) \\ &= P_{i,0}^{-1}(1|0)(\hat{x}_{i,\rho-1}(1|1) + K_{i,\rho}(1)(y_{i,\rho}(1) - C_{i,\rho}(1)\hat{x}_{i,\rho-1}(1|1)) - \hat{x}_{i,0}(1|0)) \\ &= \hat{d}_{i,\rho-1}(0) + P_{i,0}^{-1}(1|0)K_{i,\rho}(1)(y_{i,\rho}(1) - C_{i,\rho}(1)\hat{x}_{i,\rho-1}(1|1))\end{aligned}$$

which could be summarized to

$$\hat{d}_{i,\rho}(0) = \hat{d}_{i,\rho-1}(0) + C_{i,\rho}^T(1)\Psi_{i,\rho}^{-1}(1|1)(y_{i,\rho}(1) - C_{i,\rho}(1)\hat{x}_{i,\rho-1}(1|1)) \tag{3.89}$$

Based on Eq.3.83, the cost function in this case could be expressed in an alternative form as

$$J_{i,1}^{\rho}(x(1), d(0)) = J_{i,const}^{\rho}(1) + \frac{1}{2}\left\|x(1) - \hat{x}_{i,\rho}(1|1)\right\|_{P_{i,\rho}^{-1}(1|1)}^2 \tag{3.90}$$

3.3.3 During time period $k = [l+1, l+2)$

Without communication iteration

We assume that the estimation results during the time period $k = [1, l]$ are true, then the cost function $J_{i,l}^{\rho}$ is obtained by the last time instant $k = l$ as follow

$$J_{i,l}^{\rho}(x(l), d(l-1)) = \begin{pmatrix} \frac{1}{2}x^T(0)P_{i,0}^{-1}(0)x(0) + \frac{1}{2}\sum_{k=0}^{l-1} d^T(k)d(k) + \\ +\frac{1}{2}\sum_{k=0}^{l}\sum_{\xi=0}^{\rho}\Big((y_{i,\xi}(k) - C_{i,\xi}(k)x(k))^T\Sigma_{\varepsilon_{i,\xi}(k)}^{-1}(y_{i,\xi}(k) - C_{i,\xi}(k)x(k))\Big) \end{pmatrix}$$

leads to

$$J^{\rho}_{i,l}(x(l), d(l-1)) = J^{\rho}_{i,const}(l) + \frac{1}{2}\left\|x(l) - \hat{x}_{i,\rho}(l|l)\right\|^2_{P^{-1}_{i,\rho}(l|l)} \tag{3.91}$$

On time instant $k = l+1$, the cost function $J^0_{i,l+1}$ at node i is governed as

$$J^0_{i,l+1}(x(l+1), d(l)) = \begin{pmatrix} \frac{1}{2}x^T(0)P^{-1}_{i,0}(0)x(0) + \frac{1}{2}\sum_{k=0}^{l-1} d^T(k)d(k) + \frac{1}{2}d^T(l)d(l) \\ +\frac{1}{2}\sum_{k=0}^{l}\sum_{\xi=0}^{\rho}\left((y_{i,\xi}(k) - C_{i,\xi}(k)x(k))^T\Sigma^{-1}_{\varepsilon_{i,\xi}(k)}(y_{i,\xi}(k) - C_{i,\xi}(k)x(k))\right) + \\ \frac{1}{2}(y_{i,0}(l+1) - C_{i,0}(l+1)x(l+1))^T\Sigma^{-1}_{\varepsilon_{i,0}(l+1)}(y_{i,0}(l+1) - C_{i,0}(l+1)x(l+1)) \end{pmatrix}$$

The cost function could be simplified by

$$J^0_{i,l+1}(x(l+1), d(l)) = \begin{pmatrix} J^{\rho}_{i,const}(l) + \frac{1}{2}\left\|x(l) - \hat{x}_{i,\rho}(l|l)\right\|^2_{P^{-1}_{i,\rho}(l|l)} + \frac{1}{2}\left\|d(l)\right\|^2 \\ +\frac{1}{2}\left\|y_{i,0}(l+1) - C_{i,0}(l+1)x(l+1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(l+1)}} \end{pmatrix} \tag{3.92}$$

We would like to find $\hat{x}_{i,0}(l+1|l+1)$ and $\hat{d}_{i,0}(l)$ for given the data

$$y_{i,0}(0), \cdots, y_{i,\rho}(0), y_{i,0}(1), \cdots, y_{i,\rho}(1), \cdots, y_{i,0}(l), \cdots, y_{i,\rho}(l), y_{i,0}(l+1)$$

Then we solve the following optimization problem

$$J^0_{i,l+1}(\hat{x}_{i,0}(l+1|l+1), \hat{d}_{i,0}(l)) = \min_{x(l+1), d(l)} J^0_{i,l+1}(x(l+1), d(l))$$

Due to $x(l+1) = Ax(l) + d(l)$, it leads to $x(l) = A^{-1}x(l+1) - A^{-1}d(l)$. Then, we substitute the formula of $x(l)$ to Eq.3.92, and then

$$\begin{aligned} J^0_{i,l+1}(x(l+1), d(l)) = J^{\rho}_{i,const}(l) &+ \frac{1}{2}\left\|A^{-1}x(l+1) - A^{-1}d(l) - \hat{x}_{i,\rho}(l|l)\right\|^2_{P^{-1}_{i,\rho}(l|l)} \\ &+ \frac{1}{2}\left\|y_{i,0}(l+1) - C_{i,0}(l+1)x(l+1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(l+1)}} + \frac{1}{2}\left\|d(l)\right\|^2 \end{aligned}$$

By optimization, the partial derivative of the cost function with respect to disturbance d is denoted as

$$\begin{aligned} \frac{\partial J^0_{i,l+1}(x(l+1), d(l))}{\partial d(l)} &= d(l) - A^{-T}P^{-1}_{i,\rho}(l|l)(A^{-1}x(l+1) - A^{-1}d(l) - \hat{x}_{i,\rho}(l|l) \\ &= 0 \Rightarrow \hat{d}_{i,0}(l) = (I + AP_{i,\rho}(l|l)A^T)^{-1}(x(l+1) - A\hat{x}_{i,\rho}(l|l)) \end{aligned}$$

Let

$$P_{i,0}(l+1|l) = I + AP_{i,\rho}(l|l)A^T$$

so that

$$\hat{d}_{i,0}(l) = P_{i,0}^{-1}(l+1|l)(x(l+1) - A\hat{x}_{i,\rho}(l|l))$$

Leads to

$$\|x(l+1) - A\hat{x}_{i,\rho}(l|l)\|^2_{P_{i,0}^{-1}(l+1|l)} = \left\|A^{-1}x(l+1) - A^{-1}d(l) - \hat{x}_{i,\rho}(l|l)\right\|^2_{P_{i,\rho}^{-1}(l|l)} + \|d(l)\|^2$$

Taking $\|x(l+1) - A\hat{x}_{i,\rho}(l|l)\|^2_{P_{i,0}^{-1}(l+1|l)}$ into Eq.3.92, the cost function can be expressed as

$$\begin{aligned} J^0_{i,l+1}(x(l+1), d(l)) = J_{i,const} &+ \frac{1}{2}\|x(l+1) - A\hat{x}_{i,\rho}(l|l)\|^2_{P_{i,0}^{-1}(l+1|l)} \\ &+ \frac{1}{2}\|y_{i,0}(l+1) - C_{i,0}(l+1)x(l+1)\|^2_{\Sigma^{-1}_{\varepsilon_{i,0}(l+1)}} \end{aligned}$$

Now, we could find the optimal solution for $x(l+1)$ by optimization

$$\begin{aligned} \frac{\partial J^0_{i,l+1}(x(l+1), d(l))}{\partial x(l+1)} &= P_{i,0}^{-1}(l+1|l)(x(l+1) - A\hat{x}_{i,\rho}(l|l)) \\ &- C^T_{i,0}(l+1)\Sigma^{-1}_{\varepsilon_{i,0}(l+1)}(y_{i,l+1}(0) - C_{i,0}(l+1)x(l+1)) \end{aligned}$$

Consequently, we can obtain the state estimation by

$$\hat{x}_{i,0}(l+1|l+1) = \hat{x}_{i,0}(l+1|l) + K_{i,0}(l+1)(y_{i,0}(l+1) - C_{i,0}(l+1)\hat{x}_{i,0}(l+1|l)) \quad (3.93)$$

where

$$K_{i,0}(l+1) = P_{i,0}(l+1|l)C^T_{i,0}(l+1)\Psi^{-1}_{i,0}(l+1|l+1) \quad (3.94)$$

$$\Psi_{i,0}(l+1|l+1) = \Sigma_{\varepsilon_{i,0}(l+1)} + C_{i,0}(l+1)P_{i,0}(l+1|l)C^T_{i,0}(l+1) \quad (3.95)$$

$$P_{i,0}(l+1|l+1) = P_{i,0}(l+1|l) - K_{i,0}(l+1)\Psi_{i,0}(l+1|l+1)K^T_{i,0}(l+1) \quad (3.96)$$

$$\hat{x}_{i,0}(l+1|l) = A\hat{x}_{i,\rho}(l|l) \quad (3.97)$$

The residual signal could be expressed as follows

$$r_{i,0}(l+1) = y_{i,0}(l+1) - C_{i,0}(l+1)\hat{x}_{i,0}(l+1|l) \quad (3.98)$$

And for $\hat{d}_{i,0}(l)$

$$\begin{aligned}\hat{d}_{i,0}(l) &= P_{i,0}^{-1}(l+1|l)(\hat{x}_{i,0}(l+1|l+1) - A\hat{x}_{i,\rho}(l|l)) \\ &= P_{i,0}^{-1}(l+1|l)(\hat{x}_{i,0}(l+1|l+1) - \hat{x}_{i,0}(l+1|l)) \\ &= P_{i,0}^{-1}(l+1|l)K_{i,0}(l+1)(y_{i,0}(l+1) - C_{i,0}(l+1)\hat{x}_{i,0}(l+1|l))\end{aligned}$$

leads to

$$\hat{d}_{i,0}(l) = C_{i,0}^T(l+1)\Psi_{i,0}^{-1}(l+1|l+1)(y_{i,0}(l+1) - C_{i,0}(l+1)\hat{x}_{i,0}(l+1|l)) \tag{3.99}$$

Now, we can derive the cost function of node i when $k = l+1$ without communication with its neighbors.

$$J_{i,l+1}^0(x(l+1), d(l)) = J_{i,const}^0(l+1) + \frac{1}{2}\left\|x(l+1) - \hat{x}_{i,0}(l+1|l+1)\right\|^2_{P_{i,0}^{-1}(l+1|l+1)} \tag{3.100}$$

When communication iteration ξ equals ρ

Finally, we consider the situation that $k = l+1$ and $\xi = \rho$. Due to Eq.3.32, the cost function of node is expressed as

$$\begin{aligned}J_{i,l+1}^\rho(x(l+1), d(l)) = &\frac{1}{2}x^T(0)P_{i,0}^{-1}(0)x(0) + \frac{1}{2}\sum_{k=0}^{l} d^T(k)d(k) \\ &+ \frac{1}{2}\sum_{k=0}^{l+1}\sum_{\xi=0}^{\rho}\left((y_{i,\xi}(k) - C_{i,\xi}(k)x(k))^T\Sigma^{-1}_{\varepsilon_{i,\xi}(k)}(y_{i,\xi}(k) - C_{i,\xi}(k)x(k))\right)\end{aligned}$$

It is derived to another interpretation as follows

$$J_{i,l+1}^\rho(x(l+1), d(l)) = \begin{pmatrix} J_{i,const} + \frac{1}{2}\left\|y_{i,\rho}(l+1) - C_{i,\rho}(l+1)x(l+1)\right\|^2_{\Sigma^{-1}_{\varepsilon_{i,\rho}(l+1)}} \\ +\frac{1}{2}\left\|x(l+1) - \hat{x}_{i,\rho-1}(l+1|l+1)\right\|^2_{P^{-1}_{i,\rho-1}(l+1|l+1)} \end{pmatrix} \tag{3.101}$$

We would like to find $\hat{x}_{i,\rho}(l+1|l+1)$ and $\hat{d}_{i,\rho}(l)$ by using the data

$$y_{i,0}(0), \cdots, y_{i,\rho}(0), y_{i,0}(1), \cdots, y_{i,\rho}(1), \cdots, y_{i,0}(l), \cdots, y_{i,\rho}(l), y_{i,0}(l+1), \cdots, y_{i,\rho}(l+1)$$

to solve the optimization problem that

$$J_{i,l+1}^\rho(\hat{x}_{i,\rho}(l+1|l+1), \hat{d}_{i,\rho}(l)) = \min_{x(l+1), d(l)} J_{i,l+1}^\rho(x(l+1), d(l))$$

The partial derivative of the cost function with respect to the state $x(l+1)$ is expressed by

$$\begin{aligned}\frac{\partial J^{\rho}_{i,l+1}(x(l+1),d(l))}{\partial x(l+1)} &= P^{-1}_{i,\rho-1}(l+1|l+1)\left(x(l+1)-\hat{x}^{-1}_{i,\rho-1}(l+1|l+1)\right)\\ &- C^{T}_{i,\rho}(l+1)\Sigma^{-1}_{\varepsilon_{i,\rho}(l+1)}(y_{i,\rho}(l+1)-C_{i,\rho}(l+1)x(l+1))\\ &= 0 \Rightarrow\end{aligned}$$

As the result, we could obtain

$$\hat{x}_{i,\rho}(l+1|l+1)=\begin{pmatrix}\hat{x}_{i,\rho-1}(l+1|l+1)+\\ K_{i,\rho}(l+1)\left(y_{i,\rho}(l+1)-C_{i,\rho}(l+1)\hat{x}_{i,\rho-1}(l+1|l+1)\right)\end{pmatrix} \tag{3.102}$$

where

$$K_{i,\rho}(l+1)=P_{i,\rho-1}(l+1|l+1)C^{T}_{i,\rho}(l+1)\Psi^{-1}_{i,\rho}(l+1|l+1) \tag{3.103}$$

$$\Psi_{i,\rho}(l+1|l+1)=\Sigma_{\varepsilon_{i,\rho}(l+1)}+C_{i,\rho}(l+1)P_{i,\rho-1}(l+1|l+1)C^{T}_{i,\rho}(l+1) \tag{3.104}$$

$$P_{i,\rho}(l+1|l+1)=P_{i,l+1}(\rho-1|\rho-1)-K_{i,\rho}(l+1)\Psi_{i,\rho}(l+1|l+1)K^{T}_{i,\rho}(l+1) \tag{3.105}$$

The residual signal could be given by

$$r_{i,\rho}(l+1)=y_{i,\rho}(l+1)-C_{i,\rho}(l+1)\hat{x}_{i,\rho-1}(l+1|l+1) \tag{3.106}$$

And we can derive $\hat{d}_{i,\rho}(l)$ as follows

$$\hat{d}_{i,\rho}(l)=\hat{d}_{i,\rho-1}(l)+K^{d}_{i,\rho}(l+1)(y_{i,\rho}(l+1)-C_{i,\rho}(l+1)\hat{x}_{i,l+1}(\rho-1|\rho-1)) \tag{3.107}$$

where

$$K^{d}_{i,\rho}(l+1)=C^{T}_{i,\rho}(l+1)\Psi^{-1}_{i,\rho}(l+1|l+1) \tag{3.108}$$

At last, we could denote the cost function of node i as

$$J^{\rho}_{i,l+1}(x(l+1),d(l))=J^{\rho}_{i,const}(l+1)+\frac{1}{2}\left\|x(l+1)-\hat{x}_{i,\rho}(l+1|l+1)\right\|^{2}_{P^{-1}_{i,\rho}(l+1|l+1)} \tag{3.109}$$

The detailed algorithm for distributed H_2 observer design is summarized in Algorithm.4 step by step.

Algorithm 4 Distributed H_2 observer in a recursive form at $k = 1, \cdots, l$

Step 1: Local measurement and estimation

for $i = 1, \cdots, M(k)$ **do**
  Each sensor obtains the local measurement $y_{i,0}(k)$;
  Set $\hat{x}_{i,0}(k|k-1) = A\hat{x}_{i,\rho}(k-1|k-1)$, $P_{i,0}(k|k-1) = I + AP_{i,\rho}(k-1|k-1)A^T$
  Get $K_{i,0}(k) = P_{i,0}(k|k-1)C_{i,0}^T(k)(\Sigma_{\varepsilon_{i,0}(k)} + C_{i,0}(k)P_{i,0}(k|k-1)C_{i,0}^T(k))^{-1}$;
  Estimate: $\hat{x}_{i,0}(k|k) = \hat{x}_{i,0}(k|k-1) + K_{i,0}(k)(y_{i,0}(k) - C_{i,0}(k)\hat{x}_{i,0}(k|k-1))$;
  Get: $\Psi_{i,0}(k|k) = \Sigma_{\varepsilon_{i,0}(k)} + C_{i,0}(k)P_{i,0}(k|k-1)C_{i,k}^T(0)$
  Estimate: $\hat{d}_{i,0}(k-1) = C_{i,0}^T(k)\Psi_{i,0}^{-1}(k|k)(y_{i,0}(k) - C_{i,0}(k)\hat{x}_{i,0}(k|k-1))$;
  Get: $P_{i,0}(k|k) = P_{i,0}(k|k-1) - K_{i,0}(k)\Psi_{i,0}(k|k)K_{i,0}^T(k)$;
  Get residual signal $r_{i,0}(k) = y_{i,0}(k) - C_{i,0}(k)\hat{x}_{i,0}(k|k-1)$, and set $\xi = 1$;
end for

Step 2: Information transmission

for $i = 1, \cdots, M(k)$ **do**
  each sensor transmits its information ($y_{i,\xi-1}(k)$, variance of noise $\Sigma_{\varepsilon_{i,\xi-1}(k)}$, Σ_{w_i} and monitoring matrix $C_{i,\xi-1}(k)$) to its 1-step neighbors;
end for

Step 3: Receive information

for $i = 1, \cdots, M(k)$ **do**
  Each node receives information from its neighbors;
  Each node counts the number of its 1-step neighbors $N_i^{(1)}(k)$;
  Compute weighting matrices $\varphi_{ii}(k)$, $\varphi_{ij}(k)$ and $\Phi_i(k)$;
end for

Step 4: Update information

for $i = 1, \cdots, M(k)$ **do**
  Calculation $y_{i,\xi}(k) = \varphi_{ii}(k)y_{i,\xi-1}(k) + \sum_{j\in\mathcal{N}_i(k)}\varphi_{ij}(k)(y_{j,\xi-1}(k) + \omega_{j,\xi}(k))$;
  Compose $\Sigma_{\mathcal{E}_{i,\xi-1}(k)}$ and $\Sigma_{\Omega_i(k)}$;
  Calculate variance matrix under the communication law $\Sigma_{\varepsilon_{i,\xi}(k)} = \Phi_i(k)\Sigma_{\mathcal{E}_{i,\xi-1}(k)}\Phi_i^T(k) + \Phi_i(k)\Sigma_{\Omega_i(k)}\Phi_i^T(k)$;
  Calculate new monitoring matrix $C_{i,\xi}(k)$ by Eq.3.29;
end for

Step 5: Update estimation

for $i = 1, \cdots, M(k)$ **do**
  Computation : $K_{i,\xi}(k) = P_{i,\xi-1}(k|k)C_{i,\xi}^T(k)(\Sigma_{\varepsilon_{i,\xi}(k)} + C_{i,\xi}(k)P_{i,\xi-1}(k|k)C_{i,\xi}^T(k))^{-1}$;
  Estimation: $\hat{x}_{i,\xi}(k|k) = \hat{x}_{i,\xi-1}(k|k) + K_{i,\xi}(k)(y_{i,\xi}(k) - C_{i,\xi}(k)\hat{x}_{i,\xi-1}(k|k))$;
  $\hat{d}_{i,\xi}(k-1) = C_{i,\xi}^T(k)\Psi_{i,\xi}^{-1}(k|k)(y_{i,\xi}(k) - C_{i,\xi}(k)\hat{x}_{i,\xi-1}(k|k))$;
  Computation: $\Psi_{i,\xi}(k|k) = \Sigma_{\varepsilon_{i,\xi}(k)} + C_{i,\xi}(k)P_{i,\xi-1}(k|k)C_{i,\xi}^T(k)$;
  Computation: $P_{i,\xi}(k|k) = P_{i,\xi-1}(k|k) - K_{i,\xi}(k)\Psi_{i,\xi}(k|k)K_{i,\xi}^T(k))$;
  Residual signal generation: $r_{i,\xi}(k) = y_{i,\xi}(k) - C_{i,\xi}(k)\hat{x}_{i,\xi-1}(k|k)$
end for

Step 6: Repeat communication iteration

Check if $\xi < \rho$, then set $\xi = \xi + 1$ and repeat from **Step 2** to **Step 6**;

Step 7: Repeat time instant increment

Check if $k < N$, then set $k = k + 1$ and repeat from **Step 1** to **Step 7**;

3.4 Distributed H_2 Observer based Fault Detection

In this section, the realization of the distributed fault detection scheme will be described.

Without loss of generality, we take sensor node i as an example. We can denote the residual signal r_i as the diagnostic signal. According to the algorithm of distributed H_2 observer, each sensor node i could generate the residual signal $r_{i,\xi}(k)$ during every communication iteration $\xi = 0, 1, \cdots, N$ and every time instant $k = 0, 1, \cdots, N$.

Based on the algorithm, we can not only obtain the residual signal, but also the variance of the residual signal $\Psi_{i,\xi}(k|k)$ at every time instant and every communication iteration. In this case, depending on the residual signal and its variance, we set the test statistic equals to

$$J_{H_2,i,\xi}(k) = r_{i,\xi}^T(k)\,\Psi_{i,\xi}^{-1}(k) r_{i,\xi}(k) \tag{3.110}$$

Since $r_{i,\xi}$ is a white Gaussian noise with $N(0, \Psi_{i,\xi}(k|k))$ distribution, the test statistic

$$J_{H_2,i,\xi}(k) \sim \chi^2(m), i = 1, \cdots, M, \xi = 0, \cdots, \rho. \tag{3.111}$$

is subject to χ^2 distribution with m degrees of freedom in a fault-free case. Consequently, we can design the proper threshold for fault detection

$$J_{H_2,th} = \chi_\alpha^2(m) \tag{3.112}$$

with α denoting the upper-bound of false alarm rate.

To this end, the detection logic can be described as

$$\begin{cases} J_{H_2,i,\xi}(k) \leqslant J_{H_2,th}, fault-free\ case \\ J_{H_2,i,\xi}(k) > J_{H_2,th}, faulty\ case \end{cases} \tag{3.113}$$

As we can see, the fault detection scheme could be implemented on any sensor node. In other words, we can achieve our goal of distributed fault detection. Moreover, each sensor node could check whether the system is faulty or not after each communication iteration. It is important to mention that the fault detection result could be achieved even obtained without communication with all sensor nodes. To some extent, only some part of the nodes in the vicinity around each fault detector is available for detection. The communication

burden and time delay could be significantly reduced in this way.

3.5 Case Study

In this section, a simulation study on a multi-sensor system is used to show the feasibility of the proposed distributed H_2 observer based fault detection approach.

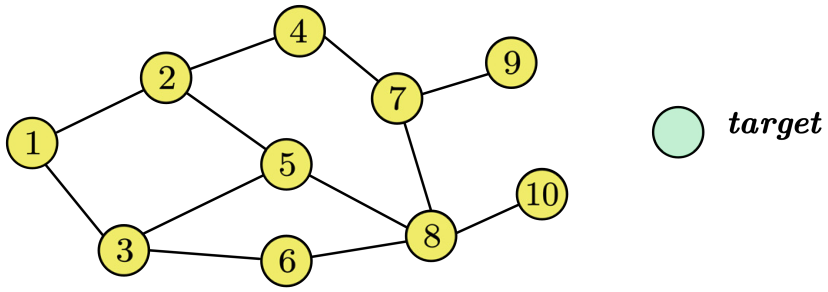

Figure 3.3: Sketch map of sensor network with target for simulation study

As shown in Fig.3.3, the green circle stands for the target, which is represented in a state-space representation as

$$x(k+1) = Ax(k) + d(k) + f(k), x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where the state vector x consists of X-axis and Y-axis position. d denotes the disturbance on the target, which is under Gaussian distribution.

$$A = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.96 \end{bmatrix}, \; d \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; \begin{bmatrix} 1E-6 & 0 \\ 0 & 1E-6 \end{bmatrix} \right)$$

And f is the fault signal for the target system, which is described by

$$f(k) = \begin{cases} \begin{bmatrix} 0 & 0 \end{bmatrix}^T & k < k_f \\ \begin{bmatrix} 1E-3 & 2E-3 \end{bmatrix}^T & k \geqslant k_f \end{cases}$$

where $k_f = 600$ denotes the fault instant of target plant.

The ten-sensor monitoring system with a graph $\mathcal{G}$ as shown in Fig.3.3 consists 10

sensor nodes. Each sensor i could obtain the local measurement of the target. It follows

$$y_i(k) = Cx_i(k) + \nu_i(k)$$

where C is the monitoring matrix, and ν denotes the vector of measurement noise, which is also under zeros mean Gaussian distribution. We assume all the sensor nodes are the same. It is worth emphasizing that the low quality of the sensors leads to the difficulty of accurate state estimation and fault detection, depending on a single sensor measurement.

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \nu \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \begin{bmatrix} 5E-2 & 0 \\ 0 & 5E-2 \end{bmatrix} \right)$$

Fig.3.4 depicts the timetable of simulation study. It is clear that the simulation of online detection lasts 800 seconds in total. The first 600 seconds of the online period are designated as an fault-free period, whereas the next 600 to 800 seconds are designated as the faulty case. A fault signal is applied on the target system, which is shown in Fig.3.3.

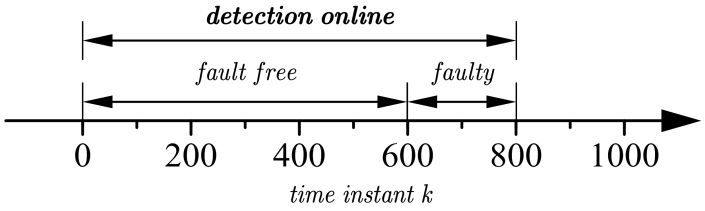

Figure 3.4: Sketch of time table for the simulation study

Without loss of generality, we take sensor node 5 as an example. The residual signal $r_{5,0}$ is a white Gaussian noise with $N(0, \Psi_{5,0}(k))$ distribution, the test statistic

$$J_{5,0}(k) \sim \chi^2(2) \tag{3.114}$$

is subject to χ^2 distribution with 2 degrees of freedom in a fault-free case. Consequently, we can design the proper threshold for fault detection

$$J_{H_2,th} = \chi^2_\alpha(2) \tag{3.115}$$

with α denoting the upper-bound of false alarm rate, here we choose $\alpha = 5\%$.

The detection results for the $5th$ nodes is sketched in Fig.3.5. After the faulty instant k_f, the test statistics generated by node 5 only depending on local measurement,

could NOT detect the faulty case of the target plant.

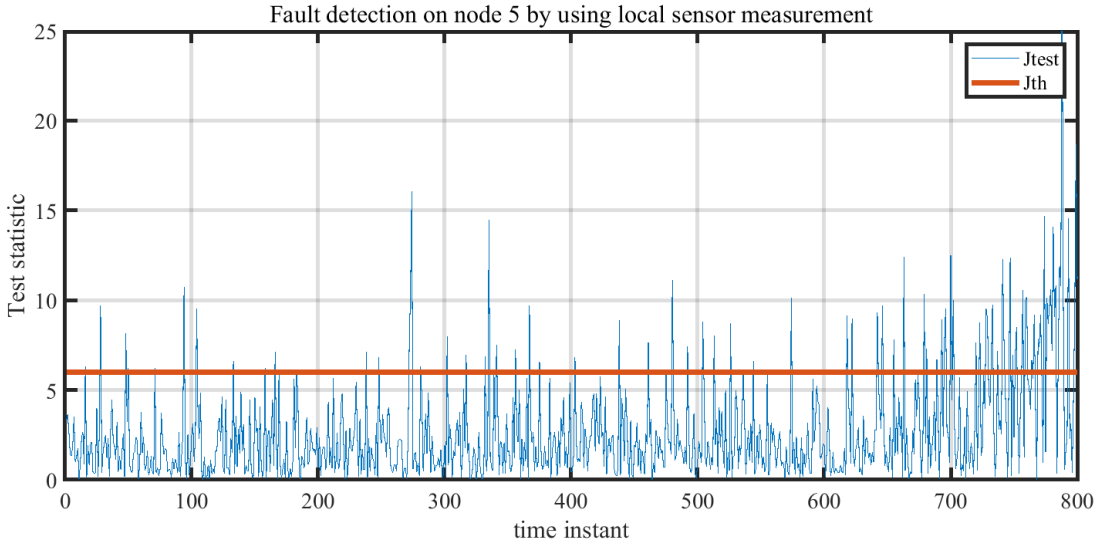

Figure 3.5: Fault detection result on node 5 by using only local measurement

To achieve target state estimation by using distributed H_2 observer, the measurement signals obtained by each sensor node transmitted through the sensor network. For simplicity, the variance of communication noise for data transmission are assumed as the same, i.e. $\Sigma_w = \begin{bmatrix} 1e-6 & 0 \\ 0 & 1e-6 \end{bmatrix}$.

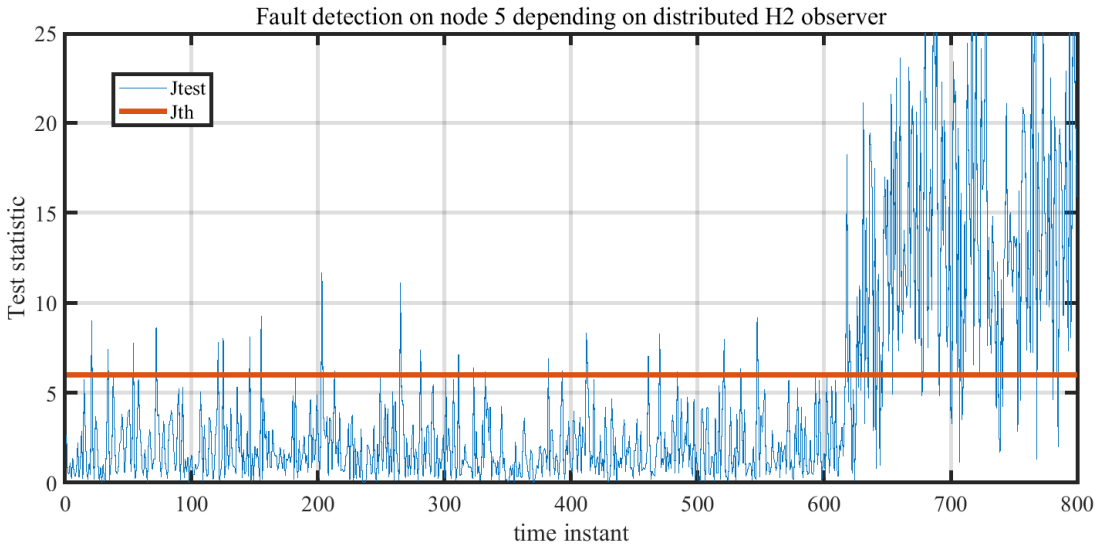

Figure 3.6: Fault detection result on node 5 depending on distributed H_2 observer

We design the weighting matrix $\Phi(k)$ as follows:

$$\Phi(k) = \{\varphi_{ij}(k)\} \in \mathcal{R}^{M(k)\times M(k)} = \begin{cases} \varphi_{ii}(k) = \frac{1}{N_i^{(1)}(k)} \\ \varphi_{ij}(k) = \varphi_{ii}(k), j \in \mathcal{N}_i^{(1)}(k) \\ \varphi_{ij}(k) = 0, j \notin \mathcal{N}_i^{(1)}(k) \end{cases} \tag{3.116}$$

Based on the distributed H_2 observer, the test statics on the $5th$ sensor node after 2-time communication iteration is also under χ^2 distribution with 2 degrees of freedom. So the fault detection threshold is the same as $J_{H_2,th} = \chi^2_\alpha(2)$, where the FAR is set as 5%.

Finally, Fig.3.6 shows the detection result obtained by the $5th$ node after a 2-time communication iteration. After the fault signal is added to the target, the test statistic generated by node 5 is significantly larger than its corresponding threshold. As a result, the simulation results can validate the effectiveness and feasibility of the detection algorithm proposed in this chapter. It is worth mentioning that each sensor node could detect the fault on the target distributively in the range of 2-step neighborhoods, even if the sensors involve large variance of measurement noises.

3.6 Concluding Remarks

In this chapter, we have dealt with state estimation and small fault detection problems by using large-scale time-varying sensor networks. The challenges of our study could be summarized from three different viewpoints: The fault detection observer should be designed with online implementation distributively; Data communication leads to a better estimation performance to fight against a high variance of local measurement noise; The communication iteration times should be limited.

First of all, models of target plant and sensor networks are established. Secondly, a communication model for data transmission between sensor nodes is designed in a recursive form. Thirdly, a distributed H_2 observer is designed in a recursive form, embedded on each node of time-varying scalable sensor networks. Furthermore, a distributed detection scheme on each sensor is proposed. Finally, the performance of estimation and fault detection depending on a distributed H_2 observer are verified by a simulation case study.

4 Distributed detection of DoS attacks on MASs

MASs can be used to construct sensor network systems of autonomous distributed mobile sensing agents, to offer a wide range of applications in decision making, data fusion, and transmission through wireless networks. To put it another way, wireless sensor networks (WSNs) can be regarded as a special case of MASs. Because of the weaknesses in WSNs, such agents are known as intelligent sensor nodes, which are vulnerable to most security attacks.

Denial-of-Service (DoS) attack is one of the most popular attacks against WSNs. The primary goal of DoS is to cause service interruption by attempting to restrict access to a service rather than compromising the service itself. By focusing on either the network's bandwidth or its connection, this type of attacks seeks to render a network unable to provide normal service [87, 63].

To this end, if only a portion of the communication channels is destroyed by adversaries, some tasks will be failure. For example, in the case of average consensus-based state estimation, weighting matrices of consensus methods are designed offline depending on the graph of a network. After a topological change, certain disconnections between numerous paired nodes will lead to a low convergence performance, and even the consensus process malfunctioning. Moreover, because most distributed observers are designed offline without taking topological changes into account, the gain matrix of a distributed observer in a sensor network should be computed online considering changes in the related laplacian matrix, or the dynamics of the distributed observer would be unstable.

Assumption 4.1. *In this chapter, the scenarios where DoS attacks have made the whole network inoperable are ignored. We only devote ourselves to studying the situation in that only a part of the communication links are disrupted, but the attacked graph of network systems remains connected.*

To detect such kinds of DoS attacks, we first introduce a wireless sensor network for

monitoring a dynamic process and present the problem formulation. Next, a Kalman-filter-based diagnostic signal generation method is proposed, which is operated distributively at each sensor node. Based on the statistical analysis of the diagnostic signal, the test statistic evaluation function and decision logic are studied. After that, a generalized likelihood ratio (GLR) method-based online detection algorithm is proposed to handle the detection problem, combined with an offline statistical training method to compute the corresponding threshold. Finally, in order to verify the feasibility of the proposed detection algorithm, we provide a straightforward simulation study on a sensor network.

4.1 System Configuration and Problem Formulation

In this chapter, we study a scenario as shown schematically in Figure 4.1 that a wireless sensor network with M sensor nodes monitors a dynamic process (marked Target in red). It should be emphasized that the goal of our work is to determine if the network system has been disrupted by DoS cyber-attacks, rather than fault detection of the target dynamic process. To this end, it is supposed a discrete LTI state-space representation with model uncertainties as follows:

$$x(k+1) = Ax(k) + \omega(k) \tag{4.1}$$

where $x \in \mathcal{R}^{n_x}$ indicates the state vector of the process under consideration. Similar to our early study, $\omega \sim N(0, \Sigma_\omega)$ denotes the process noise vector, which is supposed as zero mean white Gaussian process. And A represents the system matrix of the dynamic process.

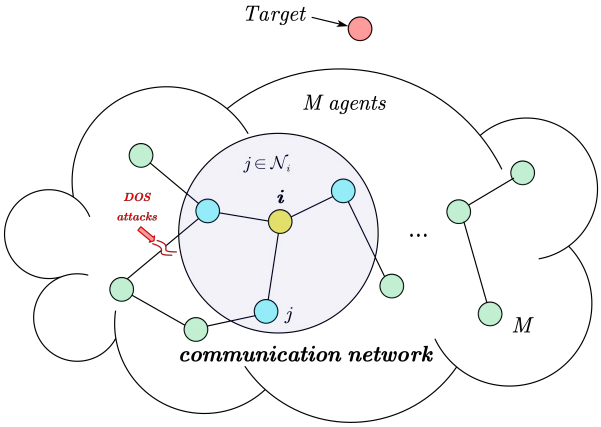

Figure 4.1: A sensor network consisting of M nodes under DoS attacks

For a dynamic process monitoring, the model of sensor node i is described by:

$$y_i(k) = C_i x(k) + \nu_i(k) \in \mathcal{R}^{n_y}, i = 1, \cdots, M. \tag{4.2}$$

where $y_i(k)$ denotes the measurement vector with n_y dimension. The measurement matrix of node i is represented as $C_i \in \mathcal{R}^{n_y \times n_x}$ and can be considered to be differ on various sensor nodes. $\nu_i(k)$ is the measurement noise of sensor i, which is assumed to be

$$\nu_i(k) \sim N(0, \Sigma_{v,i}), \Sigma_{v,i} > 0, \mathcal{E}\left(\nu_i \nu_j^T\right) = \begin{cases} \Sigma_{\nu,i}, i = j \\ 0, i \neq j \end{cases} \tag{4.3}$$

and uncorrelated with $x(k)$ and $\omega(k)$. And the variance of measurement noise is supposed to vary among different sensor nodes.

Assumption 4.2. *For the sake of simplicity, we assume that the sampling time at different sensor nodes or it should be the same, and ignore the problems of the time-inconsistency and time-delay for the wireless sensor network.*

The M sensor nodes communicate with one another via wireless data transmission, forming a sensor network with high measurement redundancy that can be used for more precise measurement of process variables, more accurate estimation of process states, and more reliable detection of faults.

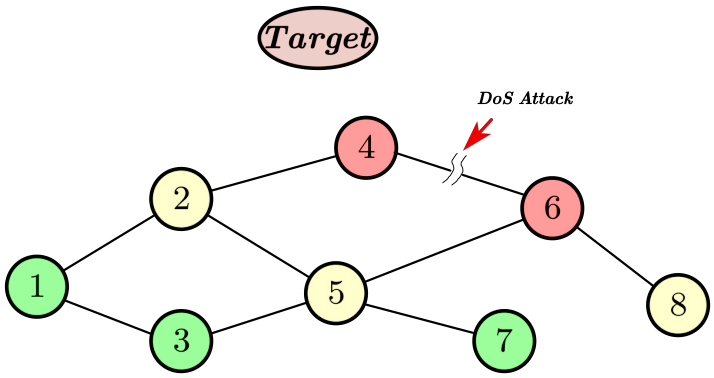

Figure 4.2: Sketched graph of a simple case under DoS attack

When sensor networks suffer from DoS attacks, as sketched in Figure 4.1, one or more communication links between sensor nodes are interrupted by adversaries. In this instance, the corresponding laplacian and adjacent matrices of sensor networks under attack are represented as $\mathcal{L}_f$ and $\mathcal{A}_f$, respectively. We define $\mathcal{L}_0$ and $\mathcal{A}_0$ as the laplacian and adjacent matrices in normal operation without attacks to emphasize the difference from the ones in

attacked situations.

As shown in Figure 4.2, we take a simple network system with 8 nodes as an example. In attack-free case, the corresponding laplacian matrix and adjacent matrix are described as follows:

$$\mathcal{L}_0 = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 2 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \mathcal{A}_0 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

If the link between node 4 and node 6 is disrupted through a DoS cyber-attack, the laplacian matrix and adjacent matrix of an attacked graph are represented respectively by

$$\mathcal{L}_f = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \mathcal{A}_f = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Sensor networks will be deactivated as a result of such cyber-attacks, which motivates us to propose a distributed detection scheme to fulfill the following requirements:

- without a central detector, each sensor node equipped with a local detector can determine whether the communication links between each node and its neighbors are disrupted by adversaries or not.
- the detector on node i has the ability to expand its detection range to communication links in the set of its ρ-steps neighbors.
- the diagnostic signal exchanged between two nodes should make it impossible for adversaries to identify the global system. In this instance, the residual signal may be the best option because it contains noise information and is difficult for opponents

to determine its physical meaning.

- threshold setting should be obtained offline, while the detection algorithm should be implemented on real-time applications.

4.2 Diagnostic Signal Generation

In this section, we should first provide a method to generate a diagnostic signal for each detector operating independently at each sensor node, to solve the distributed detection problem of DoS cyber-attacks on wireless sensor networks. Then, the statistical analysis of the diagnostic signal should be developed in preparation for a subsequent study on the test statistic evaluation function and decision logic.

4.2.1 Kalman filter based residual generator on each node

The well-known residual signal delivered by a Kalman filter is a wise choice as a diagnostic signal for detecting cyber-attacks on sensor network systems. One of the primary reasons is that each agent in most intelligent sensor networks is equipped with a Kalman filter so that the Kalman filter-based residual signal is off-the-shelf and there is no need to generate additional signals. If not, a Kalman-based post-filter can be designed following a unified observer at each sensor node for process monitoring.

Based on the model description Eq.4.1 and Eq.4.2 with the assumption of noise (4.3), we propose the following Kalman filter based residual generator running at sensor node i

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + L_{K,i}(k)r_{0K,i}(k) \tag{4.4}$$

$$\hat{y}_i(k) = C_i\hat{x}_i(k) \tag{4.5}$$

$$r_{0K,i}(k) = y_i(k) - \hat{y}_i(k) \tag{4.6}$$

where $\hat{x}_i(k) \in \mathcal{R}^{n_x}$ and $r_{0K,i}(k) \in \mathcal{R}^{n_y}$ represent the estimation of process state and the residual signal, respectively, which are generated locally by the Kalman filter on node i.

Assumption 4.3. *It is assumed that each sensor node has the knowledge of the system matrix A, the covariance of process noise Σ_ω and $x(0)$, i.e. the initial state vector of the dynamic process under consideration.*

And here $L_{K,i}(k)$ in Eq.4.4 indicates the Kalman filter gain of node i, which is a time-varying vector and can be computed recursively as follows:

- setting initial conditions

$$\hat{x}_i(0) = 0, P_i(0) = x(0)x^T(0) \tag{4.7}$$

with $P_i(k)$ denoting the variance of the estimation error of node i, which holds

$$P_i(k) = e_i(k)e_i^T(k) \tag{4.8}$$
$$e_i(k) = x(k) - \hat{x}_i(k) \tag{4.9}$$

- computing covariance of residual signal

$$R_{e,i}(k) = \Sigma_{\nu,i} + C_i P_i(k) C_i^T \tag{4.10}$$

- obtaining Kalman filter gain

$$L_{K,i}(k) = AP_i(k)C_i^T R_{e,i}^{-1}(k) \tag{4.11}$$

- updating covariance of estimation error

$$P_i(k+1) = AP_i(k)A^T - L_{K,i}(k)R_{e,i}(k)L_{K,i}^T(k) + \Sigma_\omega \tag{4.12}$$

According to our previous discussion in Chapter 2, without considering disturbances and faults, $r_{0K,i}(k)$ should be a zero mean white Gaussian process.

The computation algorithm Eq.4.7 - Eq.4.12 for obtaining the Kalman filter gain can be utilized in online implementation, which is highly computation consuming. To this regards, the observer gain matrix can be determined using the (steady) Kalman filter algorithm, which is given by

$$L_{K,i} = AP_i C_i^T R_{e,i}^{-1} \tag{4.13}$$

with

$$R_{e,i} = \Sigma_{\nu,i} + C_i P_i C_i^T. \tag{4.14}$$

where P_i is a positive definite solution and can be obtained offline by solving an algebraic

Riccati equation by

$$P_i = AP_iA^T - L_{K,i}R_{e,i}L_{K,i}^T + \Sigma_\omega \tag{4.15}$$

In this case, the residual generator can be written by the following form

$$\begin{cases} \hat{x}_i(k+1) & = A\hat{x}_i(k) + L_{K,i}(y_i(k) - C_i\hat{x}_i(k)) \\ r_{0K,i}(k) & = y_i(k) - C_i\hat{x}_i(k) \end{cases} \tag{4.16}$$

Once we get the Kalman filter-based residual signal on each sensor node, the following paragraph will propose a communication algorithm for diagnostic signal generation.

4.2.2 Residual signal communication

To detect the disrupted communication links caused by DoS attacks, an appropriate diagnostic signal should be generated and delivered to the detector. Thanks to the local Kalman filter-based residual and the information exchanging channels, we propose a communication iteration algorithm for each sensor node to compute diagnostic signals by broadcasting its local residual vector $r_{0K,i}$ to its neighbors and receiving the vectors from all its neighbors. Unlike average consensus algorithms, the communication algorithm is proposed without a complex computation of weighting matrices, and is given as follows:

$$z_{i,\xi+1}(k) = \sum_{j\in\mathcal{N}_i}(z_{i,\xi}(k) - z_{j,\xi}(k) + \varepsilon_{j,\xi}(k)) \in \mathcal{R}^{n_y},\ i = 1, \cdots, M,\ j \in \mathcal{N}_i \tag{4.17}$$

with $\xi = 1, \cdots, \rho$ here indicating the number of communication iterations. According to our detection goal, ρ-steps neighborhood should be taken into consideration. $z_{i,\xi}$ is defined as the communication vector generated on node i after the ξ^{th} iteration step, and which is delivered to neighbors of node i for the $(\xi+1)^{th}$ iteration. In this case, during the time interval $[kT_s, (k+1)T_s]$, ρ iterations of data exchanging among the sensor nodes should be guaranteed, and finally the diagnostic signal $z_{i,\rho}(k)$ on node i is obtained for detecting cyber-attacks, where T_s represents the sampling time of all the sensors.

It is worth noting that the initial condition $z_{i,0}(k)$ holds that

$$z_{i,0}(k) = r_{0K,i}(k),\ i = 1, \cdots, M \tag{4.18}$$

which equals the local Kalman filter-based residual on sensor node i at time instant k.

Furthermore, $\varepsilon_{j,\xi}(k)$ represents the communication noise during the ξ^{th} transmission from node j.

Assumption 4.4. *We assume that the communication noise is also supposed as a white Gaussian process and satisfy*

$$\varepsilon_{i,\xi} \sim N\left(0, \Sigma_{\varepsilon,i}\right), \Sigma_{\varepsilon,i} > 0,\ \mathcal{E}\left(\varepsilon_{i,\xi}\varepsilon_{j,\xi}^{T}\right) = \begin{cases} \Sigma_{\varepsilon,i}, i = j \\ 0, i \neq j \end{cases},\ i = 1, \cdots, M \tag{4.19}$$

where $\Sigma_{\varepsilon,i}$ stands for the covariance of communication noise ε_i, which indicates that the covariance remains constant during data transmission from node i. Furthermore, it is assumed that communication noise is unrelated to measurement noises and process noises.

Notably, another interpretation of the communication iteration algorithm Eq.4.17 for the wireless sensor network is that, each sensor node gathers communication output vectors from all of its neighbors and then sends the communication vector back to them after the iteration calculation. When $\xi = 0$, the communication output vector is the Kalman filter-based residual. The iteration with increasing ξ will continue until $\xi = \rho$ is achieved.

With the help of the laplacian matrix $\mathcal{L}$ of the graph theory, we can rewrite the communication algorithm (4.17) as follows:

$$z_{i,\xi+1}(k) = \mathcal{L}_{ii} z_{i,\xi}(k) + \sum_{j \in \mathcal{N}_i} \mathcal{L}_{ij} z_{j,\xi}(k) + \sum_{j \in \mathcal{N}_i} a_{ij} \varepsilon_{j,\xi}(k) \tag{4.20}$$

And the diagnostic signal $z_{i,\rho}(k)$ for each node i at time instant k can be derived by

$$z_{i,\rho}(k) = \mathcal{L}_{ii} z_{i,\rho-1}(k) + \sum_{j \in \mathcal{N}_i} \mathcal{L}_{ij} z_{j,\rho-1}(k) + \sum_{j \in \mathcal{N}_i} a_{ij} \varepsilon_{j,\rho-1}(k) \tag{4.21}$$

Now, we integrate the communication algorithm for all nodes into a compact model of an entire wireless sensor network as follows:

$$Z_{\xi+1}(k) = \tilde{\mathcal{L}} Z_{\xi}(k) + \tilde{\mathcal{A}} \mathcal{E}_{\xi}(k) \tag{4.22}$$

where $Z_{\xi}(k)$ and $\mathcal{E}_{\xi}(k)$ denote the compact vectors of communication output and communication noise, respectively, at the ξ^{th} iteration of all the nodes over time period

$[kT_s, (k+1)T_s]$.

$$Z_\xi(k) = \begin{bmatrix} z_{1,\xi}(k) \\ z_{2,\xi}(k) \\ \vdots \\ z_{M,\xi}(k) \end{bmatrix} \in \mathcal{R}^{Mn_y},\ \mathcal{E}_\xi(k) = \begin{bmatrix} \varepsilon_{1,\xi}(k) \\ \varepsilon_{2,\xi}(k) \\ \vdots \\ \varepsilon_{M,\xi}(k) \end{bmatrix} \in \mathcal{R}^{Mn_y}$$

We derive the extended laplacian matrix $\tilde{\mathcal{L}}$ and the extended adjacent matrix $\tilde{\mathcal{A}}$ with the aid of an identity matrix and Kronecker product.

$$\tilde{\mathcal{L}} = \mathcal{L} \otimes I_{n_y} = \begin{bmatrix} \mathcal{L}_{11} I_{n_y} & \cdots & \mathcal{L}_{1M} I_{n_y} \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{M1} I_{n_y} & \cdots & \mathcal{L}_{MM} I_{n_y} \end{bmatrix} \in \mathcal{R}^{Mn_y \times Mn_y} \tag{4.23}$$

$$\tilde{\mathcal{A}} = \mathcal{A} \otimes I_{n_y} = \begin{bmatrix} a_{11} I_{n_y} & \cdots & a_{1M} I_{n_y} \\ \vdots & \ddots & \vdots \\ a_{M1} I_{n_y} & \cdots & a_{MM} I_{n_y} \end{bmatrix} \in \mathcal{R}^{Mn_y \times Mn_y} \tag{4.24}$$

It is clear that Eq.4.22 can be given in an iteration form by

$$\begin{aligned} Z_{\xi+1}(k) &= \tilde{\mathcal{L}} Z_\xi(k) + \tilde{\mathcal{A}} \mathcal{E}_{\xi+1}(k) \\ &= \tilde{\mathcal{L}}(\tilde{\mathcal{L}} Z_{\xi-1}(k) + \tilde{\mathcal{A}} \mathcal{E}_\xi(k)) + \tilde{\mathcal{A}} \mathcal{E}_{\xi+1}(k) \\ &\cdots \\ &= \tilde{\mathcal{L}}^{\xi+1} Z_0(k) + \tilde{\mathcal{L}}^{\xi} \tilde{\mathcal{A}} \mathcal{E}_1(k) + \tilde{\mathcal{L}}^{\xi-1} \tilde{\mathcal{A}} \mathcal{E}_2(k) + \cdots + \tilde{\mathcal{L}} \tilde{\mathcal{A}} \mathcal{E}_\xi(k) + \tilde{\mathcal{A}} \mathcal{E}_{\xi+1}(k) \end{aligned}$$

We can summarize it as follows:

$$Z_{\xi+1}(k) = \tilde{\mathcal{L}}^{\xi+1} Z_0(k) + \sum_{\eta=1}^{\xi+1} \tilde{\mathcal{L}}^{\xi+1-\eta} \tilde{\mathcal{A}} \mathcal{E}_\eta(k) \tag{4.25}$$

where initial vector $Z_0(k) = \begin{bmatrix} r_{0K,1}^T(k) & r_{0K,2}^T(k) & \cdots & r_{0K,M}^T(k) \end{bmatrix}^T$ is the vector of Kalman-filter residuals for all sensor nodes, as well as the initial condition for communication iteration beginning at time instant k.

According to Eq.4.25, we further describe the diagnostic vector for all the sensor

nodes by

$$Z_\rho(k) = \tilde{\mathcal{L}}^\rho Z_0(k) + \sum_{\eta=1}^{\rho} \tilde{\mathcal{L}}^{\rho-\eta} \tilde{\mathcal{A}} \mathcal{E}_\eta(k) \tag{4.26}$$

$$= \left[z_{1,\rho}^T(k) \quad z_{2,\rho}^T(k) \quad \cdots \quad z_{M,\rho}^T(k) \right]^T \tag{4.27}$$

with ρ-steps iteration for the purpose of DoS attacks detection.

The mean value and covariance of the communication signal $Z_\rho(k)$ should be analyzed in the following subsection in both attack-free and attacked situations.

4.2.3 Diagnostic signal analysis

Now, we tend to investigate the mean value and covariance of the diagnostic vector $Z_\rho(k)$ in attack-free situation. It is of particular note that the subscript 0 in this notation indicates attack-free cases.

Firstly, the mean vector $\bar{Z}_{\rho,0}(k)$ is described as follows:

$$\bar{Z}_{\rho,0}(k) = \tilde{\mathcal{L}}_0^\rho \bar{Z}_0(k) = 0 \in \mathcal{R}^{Mn_y} \tag{4.28}$$

where $\tilde{\mathcal{L}}_0^\rho$ denotes the laplacian matrix in attack-free case. As we mentioned, the residual delivered by the local Kalman filter is zero mean. Moreover, the mean of communication noise vector is equivalent to zero, since we assume the communication noises are zero-mean white Gaussian processes. To this end, the mean value of the diagnostic vector also equals zero vector with $\mathcal{R}^{Mn_y}$ dimension.

Next, the covariance matrix $\Sigma_{Z_{\rho,0}}$ at the ρ^{th} iteration are then derived by

$$\begin{aligned} \Sigma_{Z_{\rho,0}} &= E\left(Z_{\rho,0}(k) - \bar{Z}_{\rho,0}(k))(Z_{\rho,0}(k) - \bar{Z}_{\rho,0}(k)\right)^T \\ &= \tilde{\mathcal{L}}_0^\rho \Sigma_R \left(\tilde{\mathcal{L}}_0^\rho\right)^T + \sum_{\eta=1}^{\rho} \tilde{\mathcal{L}}_0^{\rho-\eta} \tilde{\mathcal{A}}_0 \Sigma_{\mathcal{E}} \tilde{\mathcal{A}}_0^T \left(\tilde{\mathcal{L}}_0^{\rho-\eta}\right)^T \end{aligned} \tag{4.29}$$

here, $\tilde{\mathcal{A}}_0$ is the adjacent matrix in attack-free case. Σ_R represents the constant compact variance matrix of all residual vectors at time instant k while P_i is a constant matrix,

which holds

$$\Sigma_R = \begin{bmatrix} R_{e,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{e,M} \end{bmatrix} = \begin{bmatrix} \Sigma_{\nu,i} + C_1 P_1 C_1^T & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{\nu,M} + C_M P_M C_M^T \end{bmatrix}$$

and $\Sigma_{\mathcal{E}}$ denotes the following compact variance matrix of all communication noise vectors

$$\Sigma_{\mathcal{E}} = \begin{bmatrix} \Sigma_{\mathcal{E},i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{\mathcal{E},M} \end{bmatrix} \in \mathcal{R}^{Mn_y \times Mn_y}$$

The components in the covariance matrix Σ_{Z_ρ} can be divided into two categories as

$$\Sigma_{Z_{\rho,0}} = \begin{bmatrix} \Sigma_{z_1,\rho,0} & \cdots & S_{1i,\rho,0} & \cdots & S_{1M,\rho,0} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{i1,\rho,0} & \cdots & \Sigma_{z_i,\rho,0} & \cdots & S_{iM,\rho,0} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{M1,\rho,0} & \cdots & S_{Mi,\rho,0} & \cdots & \Sigma_{z_M,\rho,0} \end{bmatrix} \in \mathcal{R}^{Mn_y \times Mn_y} \tag{4.30}$$

where $\Sigma_{z_i,\rho,0}$ is the variance matrix of diagnostic signal generated on agent i, and $S_{ij,\rho,0}$ denotes the covariance matrix between node i and j.

Then, taking into consideration DoS cyber-attacks stated in Assumption 4.1, certain communication links are broken. As a result, some components of the relevant laplacian matrix and adjacent matrix are changed. As the same with the attack-free situation, it is noteworthy that the subscript f in this format denotes cases that are being DoS attacked.

With the aid of the attacked laplacian matrix $\mathcal{L}_f$ and adjacent matrix $\mathcal{A}_f$, we can derive the global diagnostic vector $Z_{\rho,f}(k)$ under DoS attacks by

$$Z_{\rho,f}(k) = \tilde{\mathcal{L}}_f^{\rho} Z_0(k) + \sum_{\eta=1}^{\rho} \tilde{\mathcal{L}}_f^{\rho-\eta} \tilde{\mathcal{A}}_f \mathcal{E}_\eta(k) \tag{4.31}$$

Similar to the attack-free case, the mean vector of (4.31) equals to zero vector as well, which is formulated as

$$\bar{Z}_{\rho,f}(k) = \tilde{\mathcal{L}}_f^{\rho} \bar{Z}_0(k) = 0 \in \mathcal{R}^{Mn_y} \tag{4.32}$$

On the other hand, the covariance matrix of $Z_{\rho,f}(k)$ can be affected by topological changes due to DoS attacks. It holds the following form

$$\begin{aligned}\Sigma_{Z_{\rho,f}} &= E\left(Z_{\rho,f}(k) - \bar{Z}_{\rho,f}(k))(Z_{\rho,f}(k) - \bar{Z}_{\rho,f}(k)\right)^T \\ &= \tilde{\mathcal{L}}_f^{\rho}\Sigma_R\left(\tilde{\mathcal{L}}_f^{\rho}\right)^T + \sum_{\eta=1}^{\rho}\tilde{\mathcal{L}}_f^{\rho-\eta}\tilde{\mathcal{A}}_f\Sigma_{\mathcal{E}}\tilde{\mathcal{A}}_f^T\left(\tilde{\mathcal{L}}_f^{\rho-\eta}\right)^T \qquad (4.33)\end{aligned}$$

$$= \begin{bmatrix} \Sigma_{z_1,\rho,f} & \cdots & S_{1i,\rho,f} & \cdots & S_{1M,\rho,f} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{i1,\rho,f} & \cdots & \Sigma_{z_i,\rho,f} & \cdots & S_{iM,\rho,f} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{M1,\rho,f} & \cdots & S_{Mi,\rho,f} & \cdots & \Sigma_{z_M,\rho,f} \end{bmatrix} \in \mathcal{R}^{Mn_y \times Mn_y} \qquad (4.34)$$

with its detailed elements including

$$\begin{cases} \Sigma_{z_i,\rho,f}, & i = j \\ S_{ij,\rho,f}, & i \neq j \end{cases}, i, j = 1, \cdots, M$$

where $\Sigma_{z_i,\rho,f}$ and $S_{ij,\rho,0}$ denote the attacked variance and covariance matrix of diagnostic signal, respectively. Likewise, we have to underline that the subscript f signifies network systems are under DoS cyber-attacks.

As can be seen, for each sensor node $i = 1, \cdots, M$, the mean vector of the diagnostic signal $z_{i,\rho}(k)$ maintains a zero vector with topological changes from attack-free to attack cases. However, DoS attacks cause the covariance matrix to change from $\Sigma_{z_i,\rho,0}$ to $\Sigma_{z_i,\rho,f}$. So that, we propose the following two hypotheses to describe the detection problem as

$$\begin{cases} z_{i,\rho}(k) \sim \mathcal{N}(0, \Sigma_{z_i,\rho,0}) & H_0 \ (no\ change) \\ z_{i,\rho}(k) \sim \mathcal{N}(0, \Sigma_{z_i,\rho,f}) & H_1 \ (change) \end{cases}, i = 1, \cdots, M \qquad (4.35)$$

Assumption 4.5. *We assume $\Sigma_{z_i,\rho,0}$ in attack-free case is known, but $\Sigma_{z_i,\rho,f}$ in the situation of DoS attack is unknown.*

Now we can express our detection goal in an alternative way that

Remark 4.1. *We try to do distributed detection of topological changes in a ρ-step neighborhood around each sensor node $i = 1, \cdots, M$ by the diagnostic signal $z_{i,\rho}(k)$, which is computed separately on each node by iteration computation with Kalman filter-based*

residual signal z_i as the initial condition.

In our study, we only consider the diagnostic signal after the ρ^{th} step iteration. Hereafter in this chapter, $z_{i,\rho}$, $\Sigma_{z_i,\rho,0}$ and $\Sigma_{z_i,\rho,f}$ can be simplified to z_i, $\Sigma_{z_i,0}$ and $\Sigma_{z_i,f}$, respectively.

4.3 Combination of GLR Algorithm and Statistical Method to Detect DoS Attacks

In the context of DoS attack detection, likelihood ratio (LR) approaches are particularly popular. Due to the changes in the covariance matrix in our case being unknown, the GLR method is first developed to generate a test statistic for detecting DoS attacks on the change of graph's topology, i.e. the laplacian matrix and adjacent matrix. Then, we attempt to reach the threshold through an offline statistical training method in attack-free situation. Finally, we summarize an algorithm to detect the DoS attacks online.

4.3.1 Detection of changes in variance via GLR algorithm

The probability density of Gaussian vector $z_i(k)$ is defined by

$$p_{\Sigma}\left(z_i\left(k\right)\right) = \frac{1}{\sqrt{2\pi \det(\Sigma_{z_i})}} e^{-\frac{1}{2} z_i^T(k) \Sigma_{z_i}^{-1} z_i(k)} \tag{4.36}$$

As a result, we propose the (log) LR for a given vector $z_i(k)$, which satisfies

$$s\left(z_i(k)\right) = \ln \frac{p_{\Sigma_f}\left(z_i(k)\right)}{p_{\Sigma_0}\left(z_i(k)\right)} \tag{4.37}$$

$$= \frac{1}{2} \ln \frac{\det\left(\Sigma_{z_i,0}\right)}{\det\left(\Sigma_{z_i,f}\right)} + \frac{1}{2} z_i^T(k) \Sigma_{z_i,0}^{-1} z_i(k) - \frac{1}{2} z_i^T(k) \Sigma_{z_i,f}^{-1} z_i(k). \tag{4.38}$$

If n samples of $z_i(l|k),\ l = 1, \cdots, n$ are available, the summarized likelihood ratio can be expressed by

$$\begin{aligned} s_{z_i}^n\left(k\right) = & \frac{n}{2} \ln \frac{\det\left(\Sigma_{z_i,0}\right)}{\det\left(\Sigma_{z_i,f}\right)} + \frac{1}{2} \sum_{l=1}^{n} z_i^T(l|k) \Sigma_{z_i,0}^{-1} z_i(l|k) \\ & - \frac{1}{2} \sum_{l=1}^{n} z_i^T(l|k) \Sigma_{z_i,f}^{-1} z_i(l|k) \end{aligned} \tag{4.39}$$

As we assumed in Assumption 4.5, in practice, the covariance matrix under DoS cyber-attacks is unknown. The maximum likelihood estimate can be used to obtain $\Sigma_{z_i,f}$ in this case. In order to maximize the LR, the estimated covariance matrix is the solution to the

following optimization problem.

$$\max_{\Sigma_{z_i,f}(k)} \left[s_{z_i}^n(k)\right] = \max_{\Sigma_{z_i,f}(k)} \begin{bmatrix} \frac{n}{2}\ln\frac{\det(\Sigma_{z_i,0}}{\det(\Sigma_{z_i,f})} + \frac{1}{2}\sum_{l=1}^{n} z_i^T(l|k)\Sigma_{z_i,0}^{-1} z_i(l|k) \\ -\frac{1}{2}\sum_{l=1}^{n} z_i^T(l|k)\Sigma_{z_i,f}^{-1} z_i(l|k) \end{bmatrix} \tag{4.40}$$

In order to maximize the above cost function, we can get the optimal solution that

$$\hat{\Sigma}_{z_i,f}(k) = \arg\max_{\Sigma_{z_i,f}(k)} \left[s_{z_i}^n(k)\right] = \frac{1}{n}\sum_{l=1}^{n} z_i(l|k) z_i^T(l|k) \tag{4.41}$$

So we can substitute $\hat{\Sigma}_{z_i,f}(k)$ in Eq.4.40, then the maximum LR can be achieved by

$$\max_{\hat{\Sigma}_{z_i,f}(k)} \left[s_{z_i}^n(k)\right] = \max_{\Sigma_{z_i,f}(k)} \frac{1}{2}\left[n\ln\frac{\det(\Sigma_{z_i,0})}{\det(\hat{\Sigma}_{z_i,f}(k))} + \sum_{l=1}^{n} z_i^T(l|k)\Sigma_{z_i,0}^{-1}(k)\, z_i(l|k) - 1\right] \tag{4.42}$$

Now, for detecting cyber-attacks around ρ-step neighborhxiaood of node i, we define the following test statistic of its detector as the maximum LR of its diagnostic signal $z_i(k)$.

$$J_{s_{z_i}^n}(k) = \max_{\Sigma_{z_i,f}(k)} \left[s_{z_i}^n(k)\right] \tag{4.43}$$

The decision logic for node i could be defined as

$$J_{s_{z_i}^n}(k) = \begin{cases} \leq J_{th,i}(k), & attack-free,\ H_0 \\ > J_{th,i}(k), & attackcase,\ H_1 \end{cases}$$

However, $J_{s_{z_i}^n}(k)$ is not under χ^2 distribution, so the determination of threshold $J_{th,i}$ for GLR method is unavailable. In the next subsection, we try to achieve the threshold by statistic method offline.

4.3.2 Offline statistical method for threshold determination

Based on the definitions of FAR, FDR, and MDR from Chapter 2, it is evident that the core of obtaining a proper threshold for detecting faults or attacks, is to make sure the probability of $J > J_{th}$ under certain (fault) conditions. At the same time, the MDR should remain in an acceptable range.

As described by Eq.4.43 in our study case, we give a cost function $J_{s_{z_i}^n}(k)$ with a random variable $s_{z_i}^n(k)$ with the known distribution in attack-free cases. Then, we try to

formulate the problem by finding $\hat{\gamma}(z_i(k))$, i.e. an estimated maximum value of $J_{s^n_{z_i}}(k)$ as

$$\hat{\gamma}(z_i(k)) = \max\left(J_{s^n_{z_i}}(k)\right) \tag{4.44}$$

which is subject to

$$p(\hat{\gamma}(z_i(k))) = prob\left(J_{s^n_{z_i}}(k) \leq \hat{\gamma}(z_i(k))\right) \geq 1-\alpha \tag{4.45}$$

with the probability greater than $1-\delta$, i.e. the confidence lever with $\delta \in (0,1)$. And here $\alpha \in (0,1)$ is a proper FAR we defined.

During the fault-free operation, we generate n samples of $z_i(l|k)$, $l = 1, \cdots, n$, for each node i. And then we calculate the likelihood ratio $z_i(l|k)$ according to Eq.4.39. After that, we compute $J_{S^n_{z_i}}(k)$ by Eq.4.42. Repeating the above process for N times, then yields $J_{S^n_{z_i}}(k)$, $k = 1, \cdots, N$.

To determine the repeating time N, we set the false alarm rate (FAR) α and confidence level $1-\delta$, then N should meet the requirement as follows:

$$N \geq \frac{\log\frac{1}{\delta}}{\log\frac{1}{1-\alpha}} \tag{4.46}$$

We try to find a proper value of $J_{th,i}$ for each node i, so that the probability of $J_{S^n_{z_i}}(k) \leq J_{th,i}$ is larger than $1-\alpha$. The algorithm for threshold determination offline is organized below.

Algorithm 5 Statistical method for threshold determination offline

Step 1: Diagnostic data generation offline in attack free case

Generate diagnostic signal $z_i(l|k), l = 1, \cdots, n$ on each sensor node;

Step 2: Estimation $\hat{\Sigma}_{z_i,f}(k)$ and calculation $J_{S^n_{z_i}}(k)$

Estimate $\hat{\Sigma}_{z_i,f}(k)$ by using Maximum Likelihood Eq.4.41;

Calculate $J_{S^n_{z_i}}(k)$ Eq.4.43 on each sensor node;

Step 3: Threshold determination

Repeat from **Step 1** to **Step 2** by N times;

Determine $J_{th,i}$ so that $Prob(J_{S^n_{z_i}}(k) \leq J_{th,i}) > 1-\alpha$.

4.3.3 Online detection algorithm of DoS attack

Once we obtain the threshold to detection of topological change, DoS attacks around ρ-step neighborhood of each node can be detected online.

Firstly, we collect n detection data $z_i(l|k)$, $l = 1, \cdots, n$, at each node i. Then $\hat{\Sigma}_{z_i,f}(k)$ is estimated by maximum likelihood estimation to compute $J_{S^n_{z_i}}(k)$. Finally, we check whether $J_{S^n_{z_i}}(k)$ is larger than the threshold $J_{th,i}$ or not, in order to decide the topology is attacked or attack-free. We propose an algorithm for DoS attack detection online as follows:

Algorithm 6 Detection of DoS attack online

Step 1: Diagnostic data generation online

Generate $z_i(l|k)$ for n times on each sensor node;

Step 2: Estimation $\hat{\Sigma}_{z_i,f}(k)$ and calculation $J_{S^n_{z_i}}(k)$

Estimate $\hat{\Sigma}_{z_i,f}(k)$ by using Maximum Likelihood Estimation (MLE);
Calculate $J_{S^n_{z_i}}(k)$;

Step 3: Decision making

$$J_{S^n_{z_i}}(k) = \begin{cases} \leq J_{th,i}, & attack - free \\ > J_{th,i}, & attacked \end{cases}$$

4.4 Case Study

In this subsection, a simulation study on a sensor network is used to show the feasibility of the proposed GLR based detection approach of DoS attack.

A target and the sensor network with $M = 8$ sensor nodes are sketched in Figure 4.2. We give a state-space representation of the target process as Equation (4.1), where the system matrix and the variance of process noise are denoted as follows:

$$A = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.96 \end{bmatrix}, \ \Sigma_\omega = \begin{bmatrix} 1e-6 & 0 \\ 0 & 1e-6 \end{bmatrix}$$

We define $n_x = 2$ in Equation (4.1) and the sampling time as 1 second. The initial state vector of the target process is set as $x_0 = \begin{bmatrix} 0.8 & 0.6 \end{bmatrix}^T$. As described as Equation (4.3), the

measurement matrices of each sensor node i is described as

$$C_i \in \mathcal{R}^{2\times 2} = \begin{bmatrix} i & 0 \\ 0 & M-i \end{bmatrix}, i = 1, 2, \cdots, M, \ \Sigma_\nu = \begin{bmatrix} 1e-6 & 0 \\ 0 & 1e-6 \end{bmatrix}$$

We assume that the measurement noises for all the nodes are white Gaussian process with the same variance matrix Σ_ν.

Figure 4.3 depicts the case study's timetable. The simulation lasts 800 seconds in total, divided into an offline period (which lasts for 0 to 300 sec) and an online period (300 - 800 sec). The first 300 seconds of the online period are designated as an attack-free period, whereas the next 600 to 800 seconds are designated as the attack case, in which the connection between nodes 4 and 6 is interrupted.

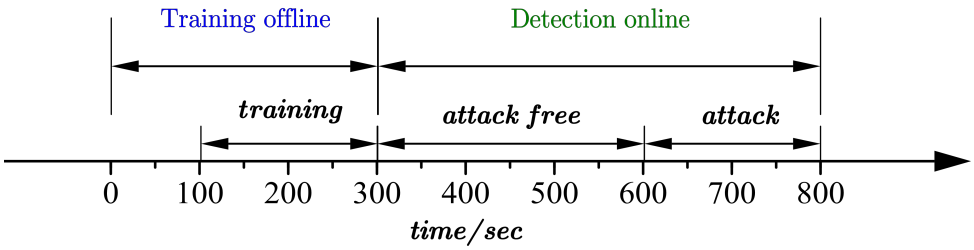

Figure 4.3: Sketch of time table for the case study

During the simulation, the residual signals for all the sensor nodes are shown in Figure 1.4. As we can see, the residual signal is also a white Gaussian process with zero mean.

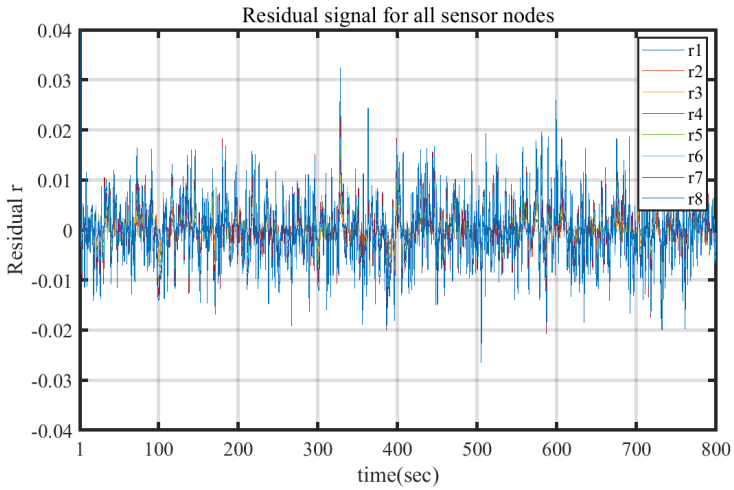

Figure 4.4: Residual signal for all sensor nodes

To compute diagnostic signal on each sensor node, the residual signals are transmitted

through sensor network. For simplicity, the variance of communication noise for each sensor are assumed as the same, i.e. $\Sigma_\varepsilon = \begin{bmatrix} 1e-6 & 0 \\ 0 & 1e-6 \end{bmatrix}$.

It is worth noting that we define the maximum iteration time ρ equals 2, which means each node could detect whether the connections around its 2-steps neighbors are disconnected or not. To be specific in our case study, the DoS attack on the link between node 4 and node 6 in Figure 4.2 should be detected not only by node 4 and 6 directly, but also by node 2, 5, and 8 after two times iteration.

To improve the detection performance, we assume the diagnostic signal in each node could be generated $n = 50$ times during one sampling time. We set the FAR as 5% and confidence level as 98%, then the repeat time N is determined as

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\alpha}} = 76.26 \tag{4.47}$$

In our simulation study, the training time includes $N = 200$ sampling time, which is larger than the minimum requirements Eq.4.47.

Based on the Algorithm 4.1, the threshold for each sensor node could be computed during training time offline, which are shown in Table 4.1.

Table 4.1: Threshold for all sensor nodes

Node Id	1	2	3	4	5	6	7	8
J_{th}	231.16	164.92	77.08	75.69	70.31	53.34	61.16	53.21

The detection results for all the nodes are sketched in Figure 4.5. After the DoS attack on the link between node 4 and node 6, the test statistics generated by node 4 and node 6 are significantly larger than its corresponding threshold, as shown in Figure 4.5 (d) and (f). Furthermore, node 2, 5 and 8 can also clearly detect the topological change caused by the DoS attack. The corresponding detection results are shown in 4.5 (b), (e) and (h), respectively. However, in Figure 4.5 (a), (c) and (g), sensor node 1, 3 and 7, which are excluded from the 2-step neighborhood of attacked nodes, could NOT detect the topological change. As the result, the simulation results can validate the effectiveness and feasibility of the detection algorithm proposed in Section 4.3. It is worth mentioning that each sensor node could detect the DoS attack distributively in the range of 2-steps neighborhoods.

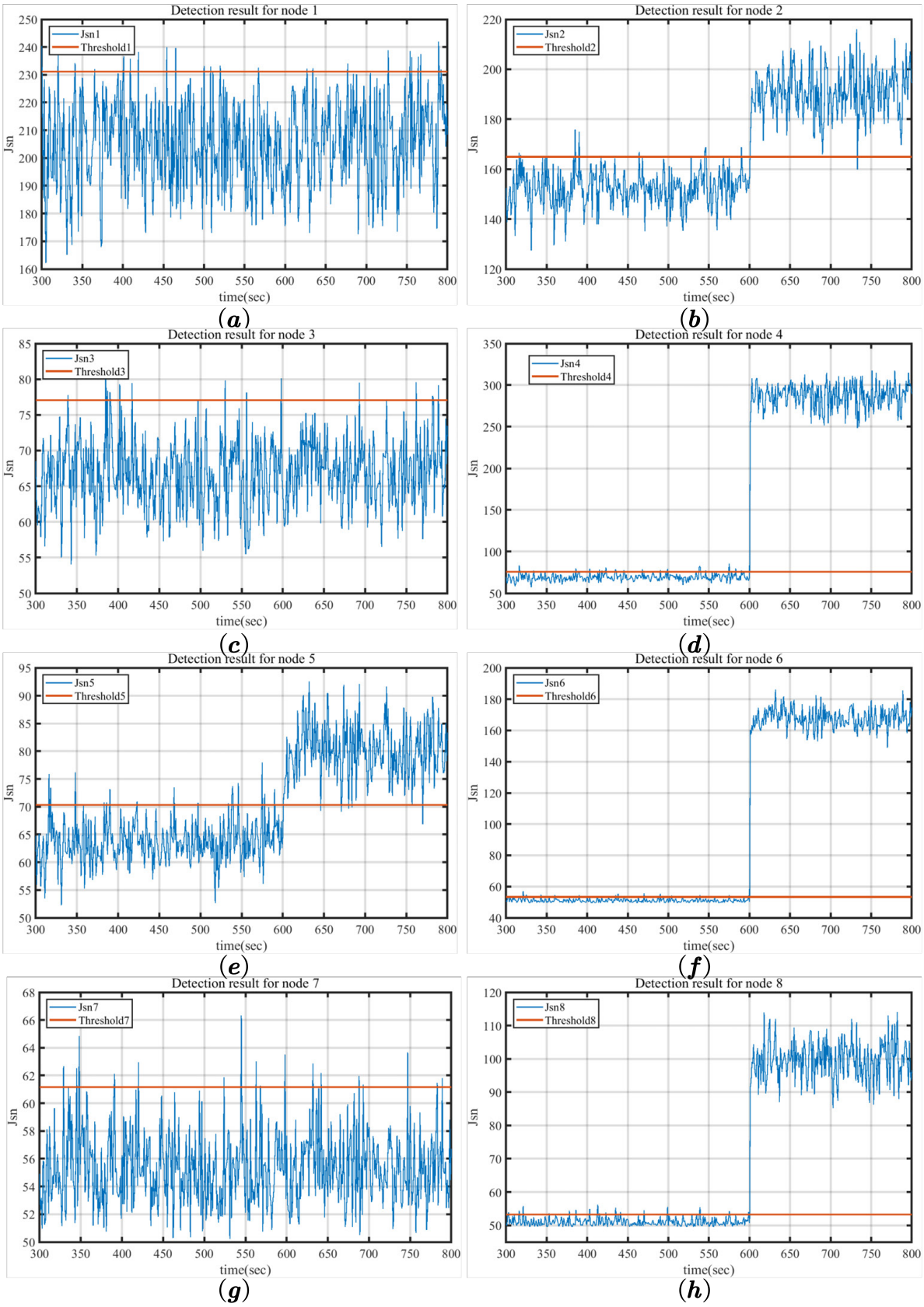

Figure 4.5: Sketch of detection results for the case study

4.5 Concluding Remarks

In this chapter, we devote ourselves to solving the distributed detection problem of DoS cyber-attacks on wireless sensor networks.

Considering the difference before and after the topological change caused by the DoS attack, we first introduce a wireless sensor network for monitoring the dynamic process of a target.

Then, based on the zero-mean residual signal generated by the local Kalman filter, we propose a communication iteration method to compute the diagnostic signal for each detector operating independently at each sensor node. Through the analysis of the diagnostic signal, its mean value does not change, and maintains zero. However, the variance of diagnosed signal varies once some communication links are disrupted by DoS attacks. As a result, we formulate the detection problem as to how we can detect changes in the variance of the diagnostic signal, instead of focusing on the detection of topological change.

After that, the GLR approach is proposed to produce a test statistic for detecting DoS attacks, since the changes in the covariance matrix in our situation are unknown. To achieve the threshold, we use an offline statistical training method in attack-free cases.

Finally, we summarize an online detection algorithm for each sensor node to detect the DoS attacks in the range of 2-step neighborhoods which are running distributively. The effectiveness and feasibility are validated by a simple simulation study.

5 Distributed Detection on Deception Attacks of MASs

This chapter deals with issuers of detecting deception cyber-attacks distributively on cooperative MASs in a consensus control configuration. As shown in Figure 5.1, a multi-agent system with M nodes interacts information among agents over a communication network. In this case, the system is extremely vulnerable to cyber-attacks.

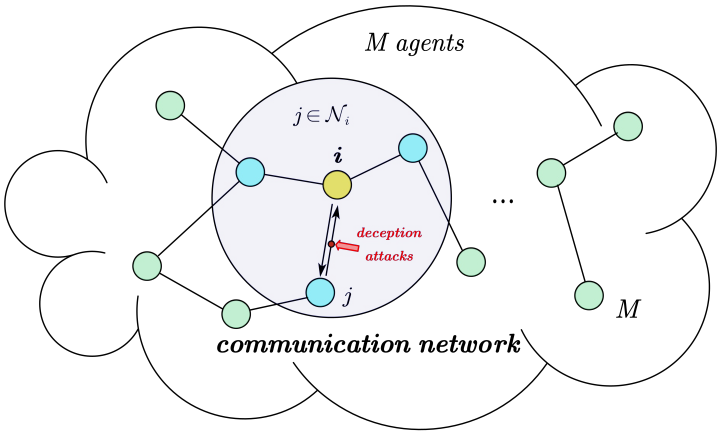

Figure 5.1: Sketch of deception attack on MASs

If some adversaries are aware of system dynamics and the functionality of existing detection methods, they can design some attack signals and inject them into cooperative MASs, which will lead to the following troubles that may considerably affect the system performance:

- if one or more agents are attacked, the performance of an entire multi-agent system is affected by communication iterations between agents, even paralyzing the entire system.
- the adversaries can design the corresponding stealthy cyber-attacks to render existing diagnostic methods ineffective.

In order to describe the problem formulation clearly, the configuration of consensus-based state feedback control on MASs considering cyber-attack is introduced first. We then

define that the false data added to a reference signals via information transmission can also be seen as stealthy cyber-attacks, which can hardly be detected by an existing standard observer-based detector. Therefore, a distributed detection scheme with an encrypted system is proposed, which allows reliable cyber-attacks detection without loss of control and monitoring performance. In addition, the detection scheme is realized on transmitting encoded signals between two agents as neighbors with each other, that prevents attacker to obtain system dynamics. Finally, the effectiveness of the distributed detection algorithm is validated by a simulation study and an experimental study on a two-robot-system.

5.1 Problem Formulation

For the purpose of a clear description about the problem formulation of distributively detecting deception attacks, in this section, we first introduce the model of a single agent considering noises. And then the single agent model is extended to a general form model of MASs. In order to achieve consensus control of MASs, a consensus-based FTC configuration is proposed. Finally, the reason why consensus cooperative MASs are vulnerable to deception cyber-attack is elaborated at the end of this section.

5.1.1 Modeling and monitoring of a single agent

Starting with a state-space model description of a single agent, we propose the system factorization and a local observer-based residual generator of a single agent in MASs.

Model description of a single agent

Given a nominal model of a single agent i

$$y_i(z) = G_i(z)\, u_i(z), y_i(z) \in \mathcal{R}^m, u_i(z) \in \mathcal{R}^p \tag{5.1}$$

where $u_i \in \mathcal{R}^{n_u}$ and $y_i \in \mathcal{R}^{n_y}$ denote the input and output vectors of the i^{th} agent, respectively. We assume that G_i is a proper real-rational matrix, so that G_i can be described in a minimal state space realization by a discrete-time LTI system as follows:

$$G_i : \begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k), x_i(0) = x_{i,0} \\ y_i(k) = C_i x_i(k) + D_i u_i(k), i = 1, \cdots, M \end{cases} \tag{5.2}$$

with $x_i \in \mathcal{R}^{n_x}$ representing the state vector, and $x_{i,0}$ denotes the initial state of the i^{th} agent. We assume A_i, B_i, C_i, D_i are system matrices of agent i, which are real constant

matrices with appropriate dimension. Combined with the process noise and measurement noise, we extend the nominal model 5.2 to the following form:

$$G_i : \begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \omega_i(k), x_i(0) = x_{i,0} \\ y_i(k) = C_i x_i(k) + D_i u_i(k) + \nu_i(k), i = 1, ..., M \end{cases} \tag{5.3}$$

here, $\omega_i(k)$ and $\nu_i(k)$ denote the process and measurement noise, respectively. We assume that both of the noises are white Gaussian processes, and satisfy the distribution $\omega_i(k) \sim N(0, \Sigma_{\omega_i})$ and $\nu_i(k) \sim N(0, \Sigma_{\nu_i})$. In addition, it is worth mentioning that the noises are statistically independent of inputs $u_i(k)$ and state vectors $x_i(k)$.

$$\mathcal{E}\left(\begin{bmatrix} \omega_i(\xi) \\ \nu_i(\xi) \\ x_i(0) \end{bmatrix} \begin{bmatrix} \omega_i(\zeta) \\ \nu_i(\zeta) \\ x_i(0) \end{bmatrix}^T\right) = \begin{bmatrix} \begin{bmatrix} \Sigma_{\omega,i} & S_i \\ S_i^T & \Sigma_{\nu,i} \end{bmatrix} \delta_{\xi\zeta} & 0 \\ 0 & \Pi_{i,0} \end{bmatrix}, \delta_{\xi\zeta} = \begin{cases} 1, \xi = \zeta, \\ 0, \xi \neq \zeta \end{cases} \tag{5.4}$$

Assumption 5.1. *For simplicity, we suppose that all agents of MASs are homogeneous in our study case, meaning that all agents share the same dynamics, system matrices, and noise distribution as well.*

To this end, the LTI representation of each agent is represented in the simple form as:

$$G_i : \begin{cases} x_i(k+1) = Ax_i(k) + Bu_i(k) + \omega_i(k), x_i(0) = x_{i,0} \\ y_i(k) = Cx_i(k) + Du_i(k) + \nu_i(k), i = 1, ..., M \end{cases} \tag{5.5}$$

And the noises for each agent i are supposed to be described as

$$\omega_i(k) \sim N(0, \Sigma_\omega), \nu_i(k) \sim N(0, \Sigma_\nu), \mathcal{E}\left(\omega_i(\xi)\nu_i^T(\zeta)\right) = S \tag{5.6}$$

System factorization of a single agent

A further system representation form of agent i could be given with a coprime factorization of a transfer function matrix over $\mathcal{RH}_\infty$. The left and right coprime factorization (LCF and RCF) of the transfer matrix $G_i(z)$ are defined by

$$G_i(z) = \hat{M}_i^{-1}(z)\hat{N}_i(z) = N_i(z)M_i^{-1}(z) \tag{5.7}$$

where the state space representation of the left and right coprime pairs (LCP and RCP) are

$$\hat{M}_i(z) = (A - LC, -L, C, I), \hat{N}_i(z) = (A - LC, B - LD, C, D) \tag{5.8}$$

$$M_i(z) = (A + BF_{0,i}, B, F_{0,i}, I), N_i(z) = (A + BF_{0,i}, B, C + DF_{0,i}, D) \tag{5.9}$$

where matrix L and $F_{0,i}$ are designed for local state estimation and state feedback control purpose, respectively, to achieve the requirements that $A - LC$ and $A + BF_{0,i}$ are Schur matrices [91, 20].

Remark 5.1. *Based on the homogeneous assumption of MASs in our study, we assume that the observer gain matrix L per agent is the same as each other. However, without loss of generality, the local feedback gain matrix $F_{0,i}$ for every agent is different because the number of neighbors per agent depends on the specific topology of multi-agent system.*

In order to hold the so-called Bezout identity

$$\begin{bmatrix} X_i(z) & Y_i(z) \\ -\hat{N}_i(z) & \hat{M}_i(z) \end{bmatrix} \begin{bmatrix} M_i(z) & -\hat{Y}_i(z) \\ N_i(z) & \hat{X}_i(z) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{5.10}$$

the corresponding RCP and LCP $\left(\hat{X}_i(z), \hat{Y}_i(z)\right)$ and $(X_i(z), Y_i(z))$ could be represented in state space formulas as

$$\hat{X}_i(z) = (A + BF_{0,i}, L, C + DF_{0,i}, I), \hat{Y}_i(z) = (A + BF_{0,i}, -L, F_{0,i}, 0) \tag{5.11}$$

$$X_i(z) = (A - LC, -(B - LD), F_{0,i}, I), Y_i(z) = (A - LC, -L, F_{0,i}, 0) \tag{5.12}$$

Observer-based residual generator of a single agent

For the purpose of local state estimation and residual generation, an observer-based residual generator is embedded locally with agent $i, i \in M$ as follows:

$$\hat{G}_i : \begin{cases} \hat{x}_i(k+1) = A\hat{x}_i(k) + Bu_i(k) + Lr_{0,i}(k) \\ \hat{y}_i(k) = C\hat{x}_i(k) + Du_i(k) \\ r_{0,i}(k) = y_i(k) - \hat{y}_i(k) \end{cases} \tag{5.13}$$

where $\hat{x}_i \in \mathcal{R}^{n_x}$ denotes the state estimate vector, and $\hat{x}_{i,0} = x_{i,0}$ is the initial condition of the state estimation for the i^{th} agent. $\hat{y}_i \in \mathcal{R}^{n_y}$ represents the output estimation vector, and $r_{0,i}$ is the primary form of a residual vector delivered by the local observer.

Based on the LCP, the local residual signal $r_{0,i}$ can be equivalently written as

$$r_{0,i}(z) = \hat{M}_i(z)\, y_i(z) - \hat{N}_i(z)\, u_i(z) \tag{5.14}$$

As we mentioned in section (2.4), if the i^{th} agent has no uncertainty, fault, and cyber-attacks, the residual signal $r_{0,i}$ can achieve zero mean. Following with the post-filter ((2.36), the local residual generator can be parameterized by

$$r_i(z) = R_i(z)\, r_{0,i}(z) = R_i(z)\,(y_i(z) - \hat{y}_i(z)) \tag{5.15}$$

where $R_i(z)$ represents the parameterization transfer function matrix of post-filter designed for detection purpose.

5.1.2 Modeling of MASs

By stacking the single agent model (5.3) considering noises, the state space realization of a multi-agent system with M agents can be derived as

$$\begin{cases} x_{\mathcal{G}}(k+1) = A_{\mathcal{G}} x_{\mathcal{G}}(k) + B_{\mathcal{G}} u_{\mathcal{G}}(k) + \omega_{\mathcal{G}}(k) \\ y_{\mathcal{G}}(k) = C_{\mathcal{G}} x_{\mathcal{G}}(k) + D_{\mathcal{G}} u_{\mathcal{G}}(k) + \nu_{\mathcal{G}}(k) \end{cases} \tag{5.16}$$

where

$$x_{\mathcal{G}}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_M(k) \end{bmatrix} \in \mathcal{R}^{n_x M}, u_{\mathcal{G}}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_M(k) \end{bmatrix} \in \mathcal{R}^{n_u M}, y_{\mathcal{G}}(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_M(k) \end{bmatrix} \in \mathcal{R}^{n_y M}$$

are the state, control and output vectors of the multi-agent system, respectively. And

$$\omega_{\mathcal{G}}(k) = \begin{bmatrix} \omega_1(k) \\ \omega_2(k) \\ \vdots \\ \omega_M(k) \end{bmatrix} \in \mathcal{R}^{n_x M}, \nu_{\mathcal{G}}(k) = \begin{bmatrix} \nu_1(k) \\ \nu_2(k) \\ \vdots \\ \nu_M(k) \end{bmatrix} \in \mathcal{R}^{n_y M}$$

represent the process noise and measurement noise vectors. Every element in the vectors is Gaussian white noise. The system matrices $A_{\mathcal{G}}$, $B_{\mathcal{G}}$, $C_{\mathcal{G}}$ and $D_{\mathcal{G}}$ of MASs are described in a diagonal form as, $A_{\mathcal{G}} = diag(A_1, \cdots, A_M) \in \mathcal{R}^{n_x M \times n_x M}$, $B_{\mathcal{G}} = diag(B_1, \cdots, B_M) \in \mathcal{R}^{n_x M \times n_u M}$, $C_{\mathcal{G}} = diag(C_1, \cdots, C_M) \in \mathcal{R}^{n_y M \times n_x M}$, and $D_{\mathcal{G}} = diag(D_1, \cdots, D_M) \in$

$\mathcal{R}^{n_y M \times n_u M}$. For the sake of simplicity, because of homogeneous assumption, the system matrices can be expressed in a compact form as follows:

$$A_{\mathcal{G}} = I_M \otimes A, \quad B_{\mathcal{G}} = I_M \otimes B,$$
$$C_{\mathcal{G}} = I_M \otimes C, \quad D_{\mathcal{G}} = I_M \otimes D.$$

with $I_M \in \mathcal{R}^{M \times M}$ denoting an identity matrix, and $\otimes$ is Kronecker product.

5.1.3 Consensus controller design of MASs

After the modelling of MASs, in this subsection, we provide two control configurations to investigate the consensus cooperation problem for MASs. For instance, output feedback consensus control architecture and consensus-based fault-tolerant controller (FTC) configuration.

Output feedback consensus controller design

In cooperative MASs, the goal of the consensus control is to achieve the average consensus of all the state vectors described in Eq.5.16, such that

$$\lim_{k \to +\infty} \|x_i(k) - x_j(k)\|^2 = 0, i, j = 1, \cdots, M \tag{5.17}$$

with a limitation that all the initial condition of state vectors should be bounded.

To meet the target, we can propose and design a simple feedback controller for each agent in the form of:

$$u_i(k) = -\sum_{j \in \mathcal{N}_i} K_{\alpha,ij}(y_i(k) - y_j(k)), i = 1, \cdots, M \tag{5.18}$$

where $K_{\alpha,ij}$ is the design parameter of output feedback controller gain matrix, depending on the coupling strength vector between agents i and j.

Assumption 5.2. *For the sake of the homogeneous assumption and simplicity, all $K_{\alpha,ij}$ for every pair of connected nodes are assumed to be identical and denoted as K_α.*

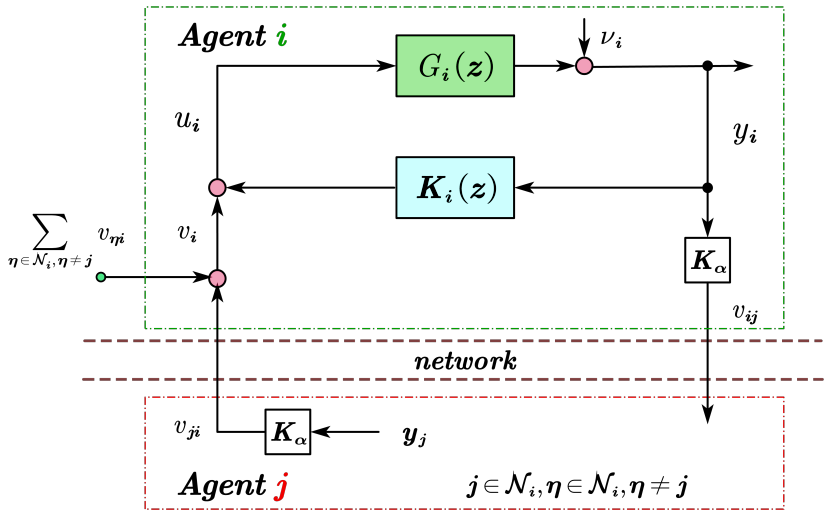

Figure 5.2: Sketch of output feedback consensus control configuration on MASs

The presentation of local output feedback control law (Eq.5.18) is able to be written as an equivalent form. As we can see in Figure 5.2, the local controller holds

$$u_i(k) = K_i y_i(k) + v_i(k), i = 1, \cdots, M \tag{5.19}$$

where K_i denotes the local output feedback gain matrix of agent i, and $v_i(k)$ can be regarded as the reference signal for the i^{th} agent. To be specific, in our study case, they are given as follows:

$$K_i = -N_i K_\alpha \tag{5.20}$$

$$v_i(k) = \sum_{j\in\mathcal{N}_i} v_{ji}(k) = \sum_{j\in\mathcal{N}_i} K_\alpha y_j(k) \tag{5.21}$$

with N_i denoting the number of the neighbors for the i^{th} agent. And $v_{ji}(k) = K_\alpha y_j(k)$ represents the reference signal generated in agent j and transmitted to agent i for the consensus control purpose.

Consensus-based fault tolerant controller design

With the increasing scale and complexity of MASs, the overall system performance or even the stability may have a higher chance of being influenced by faults, disturbances and uncertainties occurring on a single agent [11]. In recent years, FTC has been attracting more attention. The goal of FTC is to design control systems that can withstand potential failures to improve system reliability and utilization while maintaining ideal performance [66].

Therefore, in this subsection, we propose an observer-based fault-tolerant consensus controller, as shown in Figure 5.3, to mainly deal with model uncertainties and disturbances problems of a single agent.

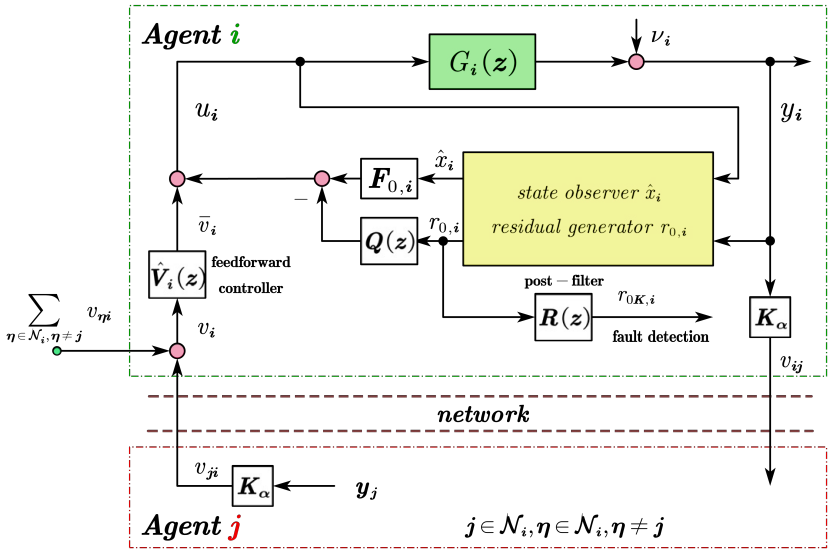

Figure 5.3: Sketch of observer-based fault tolerant consensus control configuration on MASs

According to the fault-tolerant control configuration we mentioned in Subsection 2.4, the stabilizing local output feedback control law (Eq.5.19) could be equivalently written in a form of an observer-based feedback control architecture driven by the local observer-based residual generator (Eq.5.13). It yields

$$\hat{x}_i(k+1) = (A - LC)\hat{x}_i(k) + (B - LD)u_i(k) + Ly_i(k) \tag{5.22}$$

$$r_{0,i}(k) = y_i(k) - C\hat{x}_i(k) - Du_i(k) \tag{5.23}$$

$$u_i(z) = F_{0,i}\hat{x}_i(z) - Q_i(z)r_{0,i}(z) + \bar{v}_i(z) \tag{5.24}$$

$$\bar{v}_i(z) = \left(X_i(z) - Q_i(z)\hat{N}_i(z)\right)v_i(z) \tag{5.25}$$

where $F_{0,i}$ denotes the stable observer-based state feedback gain of the i^{th} agent. And the parameterization transfer function $Q_i(z)$ is used to enhance system robustness and disturbance resistance on a single agent. We design the parameterization matrix $Q_i(z)$ offline, so that the FTC architecture could be regarded as passive fault tolerant controller (PFTC) [16]. For the sake of simplicity, the $Q_i(z)$ is set identical as $Q(z)$ for all the agents due to the homogeneous assumption.

The control law of consensus-based FTC represented in Eq.5.24 consists of two parts, which are discussed as follows:

- a local observer-based state feedback controller

$$F_{0,i}\hat{x}_i(z) - Q_i(z)\, r_{0,i}(z)$$

  guarantees the basic control performance, such as the stability, for each agent, even if the local agent is suffering limited disturbances, faults and cyber-attacks via communication.

- a feed forward controller

$$\bar{v}_i(z) = \hat{V}_i(z) v_i(z) \tag{5.26}$$

  ensures the consensus control performance by using the reference signal transmitted from all i's neighbors and summarized locally at agent i. Where $\hat{V}_i(z)$ denotes the feed-forward gain matrix, could be represented as

$$\hat{V}_i(z) = X_i(z) - Q_i(z)\,\hat{N}_i \tag{5.27}$$

As shown in Figure 5.3, the reference signals have to be transmitted via network interaction between each pair of neighbors for the sake of consensus control purpose. Take agent i and j as an example, both of the reference signals v_{ij} and v_{ji} are extremely vulnerable to cyber-attacks by adversaries. Even worse is the case that, if the attackers have the system knowledge and a precise understanding of the consensus control functions, an algorithm that generates stealthy cyber-attack can be designed to against existing detection methods. It is therefore necessary to analysis and discuss the problem formulation of MASs under cyber-attacks in next subsection.

5.1.4 Cyber-attacks on MASs and problem formulation

Due to the consensus-based FTC configuration for MASs discussed in the last section, the network communication among agents can be attacked by adversaries. Some false data is injected into the reference signals transmitted between neighbors, so that the consensus performance of MASs is no longer guaranteed.

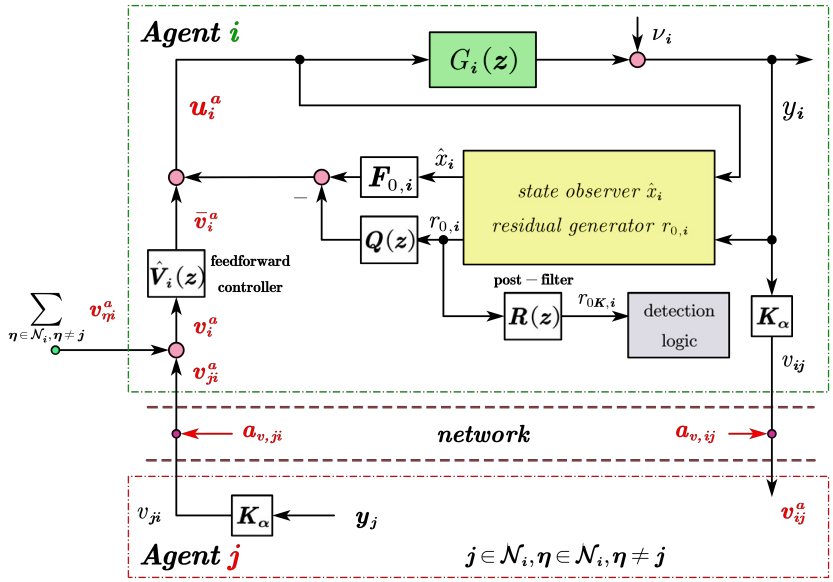

Figure 5.4: Sketch of observer-based fault tolerant consensus control configuration on MASs under cyber-attacks

For convenience, we just provide the details of cyber-attacks on agent i and its neighbors without loss of generality. Figure 5.4 obviously illustrates that, the reference signal $v_i(k)$ for agent i is integrity corrupted by attacks signals, and replaced by $v_i^a(k)$, which is the reference signal under attacks. It holds

$$v_i^a(k) = \sum_{\eta\in\mathcal{N}_i} v_{ji}^a(k), j \in \mathcal{N}_i \tag{5.28}$$

It is obvious that, besides of the reference signal v_{ji} from agent j is corrupted by the attack signal $a_{v,ji}(k)$, another attack signal $a_{v,ij}(k)$ also injects into the reference signal v_{ij} transmitted from agent i to agent j. We have

$$v_{ji}^a(k) = v_{ji}(k) + a_{v,ji}(k) \tag{5.29}$$

$$v_{ij}^a(k) = v_{ij}(k) + a_{v,ij}(k),\ j \in \mathcal{N}_i \tag{5.30}$$

The feed-forward control signal for agent i is also affected by the attack signals via the network communication with all of its neighbors. It yields

$$\bar{v}_i^a(z) = \hat{V}_i(z) v_i^a(z) \tag{5.31}$$

Combined with the attacked feed-forward control signal $\bar{v}_i^a(z)$, the control output of the consensus-based FTC could be represented as $u_i^a(k)$ instead of $u_i(k)$, which is given as

follows

$$u_i^a(z) = F_{0,i}\hat{x}_i(z) - Q_i(z)\, r_{0,i}(z) + \bar{v}_i^a(z) \tag{5.32}$$

So, by extending the state-space representation (Eq.5.3) of agent i with noises, the dynamics of the i^{th} agent considering cyber-attacks is governed by

$$G_i^a : \begin{cases} x_i(k+1) = Ax_i(k) + Bu_i^a(k) + \omega_i(k), x_i(0) = x_{i,0} \\ y_i(k) = Cx_i(k) + Du_i^a(k) + \nu_i(k), i = 1, \cdots, M \end{cases} \tag{5.33}$$

Considering cyber-attacks, the dynamics of the local standard observer-based residual generator illustrated in Figure 5.4 is rewritten as follows:

$$\hat{G}_i^a : \begin{cases} \hat{x}_i(k+1) = A\hat{x}_i(k) + Bu_i^a(k) + Lr_{0,i}(k) \\ r_{0,i}(k) = y_i(k) - \hat{y}_i(k) \\ \hat{y}_i(k) = C\hat{x}_i(k) + Du_i^a(k) \end{cases} \tag{5.34}$$

and it can be sorted into the form below:

$$\hat{x}_i(k+1) = (A - LC)\hat{x}_i(k) + (B - LD)u_i^a(k) + Ly_i(k), \tag{5.35}$$

$$r_{0,i}(k) = -C\hat{x}_i(k+1) - Du_i^a(k) + y_i(k) \tag{5.36}$$

As shown in Figure 5.4, we would like to apply the standard observer-based detector embedded on agent i to diagnose whether the reference signal for agent i is corrupted by cyber-attacks or not.

Firstly, the diagnostic signal $r_{0K,i}(k)$ is delivered by using a post-filter as follows:

$$r_{0K,i}(z) = R_{0K,i}(z)\, r_{0,i}(z) \tag{5.37}$$

here $R_{0K,i}(z)$ represents a Kalman-based post-filter [17], which holds

$$R_{0K,i}(z) = I + C(zI - A_{L_K,i})^{-1}(L - L_{K,i}), A_{L_K,i}(k) = A - L_{K,i}(k)C \tag{5.38}$$

where $L_K(k)$ represents the time-varying Kalman gain matrix, which can be determined by using the recursive Kalman filter algorithm, with the knowledge of Σ_ω, Σ_ν and S we

mentioned in Eq.5.6.

$$L_{K,i}(k) = \left(AP_i(k)C^T + S\right)\Sigma^{-1}_{r_{0K,i}}(k) \tag{5.39}$$

$$\Sigma^{-1}_{r_{0K,i}}(k) = CP_i(k)C^T + \Sigma_\nu \tag{5.40}$$

$$P_i(k+1) = AP_i(k)A^T + \Sigma_\omega - L_{K,i}(k)\Sigma_{r_{0K,i}}(k)L^T_{K,i}(k) \tag{5.41}$$

$$P_i(0) = \Pi_{i,0} = \varepsilon\left(x_i^T(0)x_i(0)\right) \tag{5.42}$$

The special post-filter with the Kalman gain matrix $L_{K,i}(k)$ devotes to generate the diagnostic signal $r_{0K,i}(k)$, which is a Gaussian white process. According to [17], $r_{0K,i}(k) \sim N(0, \Sigma_{r_{0K,i}}(k))$ with $\Sigma_{r_{0K,i}}(k)$ denoting the variance of the diagnostic signal.

Secondly, the evaluation function of diagnostic signal $r_{0K,i}(k)$ can be formulated by using χ^2 test statistic. It holds

$$J_{r_{0K,i}}(k) = r^T_{0K,i}(k)\Sigma^{-1}_{r_{0K,i}}(k)r^T_{0K,i}(k) \sim \chi^2(n_y) \tag{5.43}$$

Thirdly, the threshold J_{th} is determined according to $\chi^2_\alpha(n_y)$. Here, α denotes a given upper-bound of FAR.

To this end, we finally propose a detection logic, which is described by

$$\begin{cases} J_{r_{0K,i}}(k) \leqslant J_{th} \Longrightarrow attack - free \\ J_{r_{0K,i}}(k) > J_{th} \Longrightarrow attack\ is\ detected \end{cases}$$

Recall the state-space representations (Eq.5.33) of the attacked agent i and its local standard observer-based residual generator (Eq.5.35-Eq.5.36), the dynamics of estimation error $e_i(k)$ of agent i can be written as

$$e_i(k+1) = (A - LC)e_i(k) + \omega_i(k) - L\nu_i(k) \tag{5.44}$$

$$r_{0,i}(k) = Ce_i(k) + \nu_i(k) \tag{5.45}$$

It is worth mentioning that, related to noises only, the residual vector $r_{0,i}(z)$ is not included with any attack signals. To this end, $r_{0K,i}(k)$ is a white Gaussian noise with $N(0, \Sigma_{r_{0K,i}})$ distribution, where $\Sigma_{r_{0K,i}}$ is the variance of the residual $r_{0K,i}(k)$. Consequently, we can conclude that

Remark 5.2. *In terms of the configuration for consensus cooperative MASs as sketched in Figure 5.3, any cyber-attacks injected into the reference signals of agent* $i = 1, \cdots, M$

can not be detected by using a standard observer-based detection scheme embedded locally on the agent i.

It strongly motivates us to propose distributed cyber-attacks detection scheme on MASs. The objectives of the detector are manifolds:

- each agent i should embed a detector of cyber-attacks;
- whether the reference signal is sent to or received from a neighbor of agent i, the cyber-attacks injected into the reference signal should be detected;
- the original operation performance, such as control and monitoring performance, should not be influenced after adding the cyber-attack detector;
- the transmission signals should be encrypted with a confidential switching law to prevent the adversaries from recording information via the communication network to identify the dynamics of MASs.

5.2 Distributed Cyber-attack Detection Scheme

In order to fulfill our detection goals, in this section, an encrypted detection scheme with a switching encoder-decoder system is proposed, which is sketched schematically in Figure 5.5. In our study, we present two assumptions as follows:

Assumption 5.3. *Taking cyber-attacks on the reference signal into account, the control loop of consensus-based FTC on MASs under consideration is configured as sketched in Figure 5.4.*

Assumption 5.4. *Without loss of generality, for any agent i in a multi-agent system, both the reference signals v_{ji} and v_{ij}, no matter receiving from or sending to its neighbors $j \in \mathcal{N}_i$, are corrupted by the attack signals $a_{v,ji}$ and $a_{v,ij}$, respectively. They are described as the formulas Eq.5.30 and Eq.5.29. Due to the fact that inter-neighbor communication encompasses detection signals $u_{\sigma,ji}$, adversaries possess the capability to introduce attack signals, denoted as $a_{u,ji}$, into the detection signal, as illustrated clearly in control and detection scheme shown in Figure 5.5.*

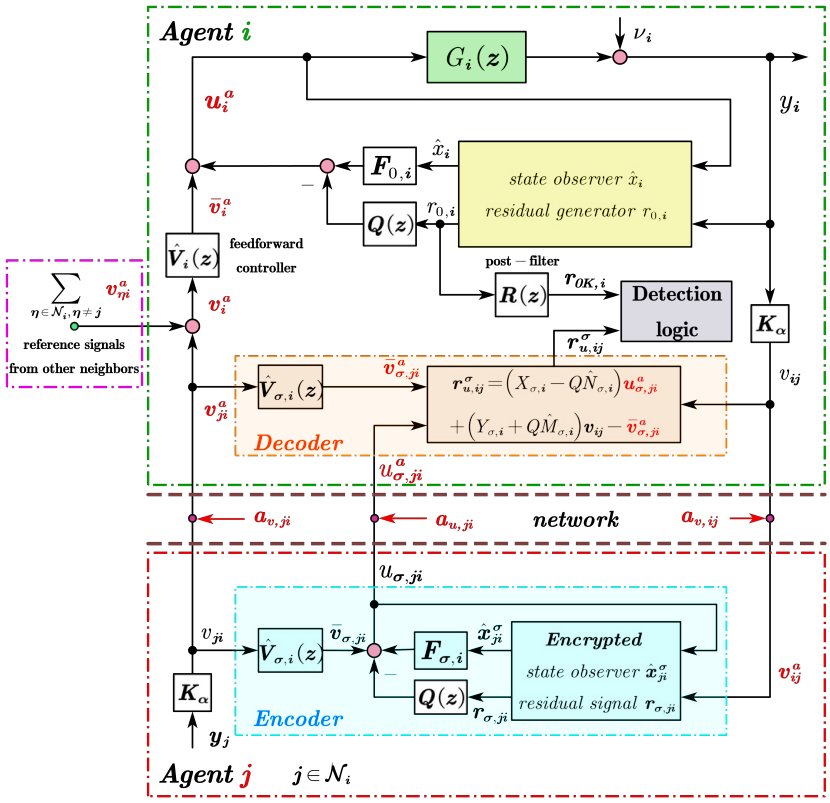

Figure 5.5: Sketch of the distributed encrypted detection scheme on MASs

5.2.1 Design and construction of the diagnostic signals

For the sake of convenience, we take any pair of two agents i and j being neighbors to each other as an example. Including an encoder and a decoder, the cyber-attack detector is distributively realized on the both sides of the two agents, i and j. To be specific, we will introduce the generation of diagnostic signals progressively.

Encoder at agent j

In order to detect cyber-attacks distributively, each agent constructs several encoders corresponding to each of its neighbor. For instance, running on the side of agent j, the encoder against agent i is shown in Figure 5.5. Similar to the FTC configuration, the encoder consists of a switching encrypted observer-based residual generator and a switching feed forward controller.

- switching encrypted observer-based residual generator

$$\begin{aligned}\hat{x}_{ji}^{\sigma}(k+1) &= A\hat{x}_{ji}^{\sigma}(k) + Bu_{\sigma,ji}(k) + L_{\sigma}\left(v_{ij}^{a}(k) - \hat{v}_{ij}^{a}(k)\right) \\ &= (A - L_{\sigma}C)\hat{x}_{ji}^{\sigma}(k) + (B - L_{\sigma}D)u_{\sigma,ji}(k) + L_{\sigma}v_{ij}^{a}(k) \qquad (5.46)\\ \hat{v}_{ij}^{a}(k) &= C\hat{x}_{ji}^{\sigma}(k) + Du_{\sigma,ji}(k) \qquad (5.47)\\ r_{\sigma,ji}(k) &= v_{ij}^{a}(k) - \hat{v}_{ij}^{a}(k) = -C\hat{x}_{ji}^{\sigma}(k) - Du_{\sigma,ji}(k) + v_{ij}^{a}(k) \qquad (5.48)\end{aligned}$$

  where $\sigma(k_s)$ denotes the switching law which is time-triggered by switching time instant k_s. Let L_σ represent a switched observer gain matrix. The reference signal $v_{ij}^{a}(k)$ can be seen as one of the inputs for the encoder. In addition, the residual signal of the encoder $r_{\sigma,ji}(k)$ is the estimation error between $v_{ij}^{a}(k)$ and $\hat{v}_{ij}^{a}(k)$.

- switching feed-forward controller

$$\begin{aligned}\bar{v}_{\sigma,ji}(z) &= \hat{V}_{\sigma,i}(z)\, v_{ji}(z) \qquad (5.49)\\ \hat{V}_{\sigma,i}(z) &= X_{\sigma,i}(z) - Q(z)\,\hat{N}_{\sigma,i}(z) \qquad (5.50)\end{aligned}$$

  Let $\hat{V}_{\sigma,i}(z)$ denote the switched feed-forward gain matrix corresponding to the i^{th} agent. And the state space realization below

$$\begin{aligned}X_{\sigma,i}(z) &= (A - L_{\sigma}C, -(B - L_{\sigma}D), F_{\sigma,i}, I)\\ Y_{\sigma,i}(z) &= (A - L_{\sigma}C, -L_{\sigma}, F_{\sigma,i}, 0)\\ \hat{M}_{\sigma,i}(z) &= (A - L_{\sigma}C, -L_{\sigma}, C, I)\\ \hat{N}_{\sigma,i}(z) &= (A - L_{\sigma}C, B - L_{\sigma}D, C, D)\end{aligned}$$

  represent the switched matrices of system factorization. Here, $F_{\sigma,i}$ is defined as the switched state feedback gain matrix and L_σ denotes the switched observer gain matrix. with σ_m denoting the maximum number of switching operation state.

Combined with the two inputs, i.e. $v_{ij}^{a}(k)$ and $v_{ji}(k)$, the encoded signal $u_{\sigma,ji}(z)$ is generated as the output of the encoder realized on agent j. It yields,

$$u_{\sigma,ji}(z) = F_{\sigma,i}\hat{x}_{ji}^{\sigma}(z) - Q(z)\, r_{\sigma,ji}(z) + \bar{v}_{\sigma,ji}(z) \qquad (5.51)$$

Recall the derivation in Chapter 2, any output feedback controller could be regarded as a observer-based state feedback controller. As the result, based on Eq.5.46, Eq.5.48 and Eq.5.51, another interpretation of $u_{\sigma,ji}(z)$ could be equivalently written as a output

feedback control form by

$$u_{\sigma,ji}(z) = K_{\sigma,i} v_{ij}^{a}(z) + v_{ji}(z) \tag{5.52}$$

where

$$K_{\sigma,i} = -\hat{V}_{\sigma,i}^{-1}(z)\,\hat{U}_{\sigma,i}(z) \tag{5.53}$$

$$\hat{U}_{\sigma,i}(z) = Y_{\sigma,i}(z) + Q(z)\,\hat{M}_{\sigma,i}(z) \tag{5.54}$$

The detailed proof is discussed in Appendix.

Notably, the differences between our encoder and the well-known FTC controller are manifolds:

Remark 5.3. *The encoded signal $u_{\sigma,ji}(z)$ is designed only for detection purpose, instead of the goal for fault-tolerant control on agent j. In this case, the control performance of agent j can not be influenced by adding the encoder.*

Remark 5.4. *Setting the parameterization matrix $Q(z)$ in Eq.5.51 is to increase complexity on the encrypted system, making it less possibility for the adversaries to identify the system.*

When it comes to the design of switching systems, the following requirements should be met:

- the switching law should be designed to satisfy the average dwell time (ADT) condition [36, 88].

- the time interval between two adjacent switching time instants should be designed such short that the attackers can not identify the switched system dynamics, i.e. $X_{\sigma,i}(z)$, $Y_{\sigma,i}(z)$, $\hat{M}_{\sigma,i}(z)$, $\hat{N}_{\sigma,i}(z)$.

- standing for the switching parameters, th switched state feedback gain matrix $F_{\sigma,i}$ and observer gain matrix L_σ should be designed to ensure that $A + BF_{\sigma,i}$ and $A - L_\sigma C$ are the Schur matrices.

After generated by the encoder in agent j, the encoded signal $u_{\sigma,ji}(z)$ will be transmitted to the corresponding agent i for the sake of cyber-attack detection.

Decoder on agent i

Figure 5.5 shows clearly that, the encoded signal $u_{\sigma,ji}(z)$ is corrupted attack signal $a_{u,ji}$ by adversaries via network communication. It gives

$$u^a_{\sigma,ji}(k) = u_{\sigma,ji}(k) + a_{u,ji}(k),\ i = 1, \cdots, M,\ j \in \mathcal{N}_i \tag{5.55}$$

Besides $u^a_{\sigma,ji}(k)$, i.e. the encoded signal form agent j, the other two inputs of the decoder realized on agent i are $v_{ij}(k)$ and $v^a_{ji}(k)$. In order to generate the diagnostic signal $r^a_{u,ij}(k)$, the decoder corresponding to agent j is described by

$$r^\sigma_{u,ij}(z) = \hat{V}_{\sigma,i}(z)\, u^a_{\sigma,ji}(z) + \hat{U}_{\sigma,i}(z)\, v_{ij}(z) - \bar{v}^a_{\sigma,ji}(z) \tag{5.56}$$

where

$$\bar{v}^a_{\sigma,ji}(z) = \hat{V}_{\sigma,i}(z)\, v^a_{ji}(z) \tag{5.57}$$

Considering the attack signals, the decoder (Eq.5.56) follows from Eq.5.55, Eq.5.30 and Eq.5.29 that

$$\begin{aligned} r^\sigma_{u,ij}(z) = {} & \hat{V}_{\sigma,i}(z)\left(u_{\sigma,ji}(z) + a_{u,ji}(z)\right) + \hat{U}_{\sigma,i}(z)\left(v^a_{ij}(z) - a_{v,ij}(z)\right) \\ & - \hat{V}_{\sigma,i}(z)\left(v_{ji}(z) + a_{v,ji}(z)\right) \end{aligned} \tag{5.58}$$

Then, substituting Eq.5.52 and Eq.5.53 into the decoder (Eq.5.58), it yields

$$\begin{aligned} r^\sigma_{u,ij}(z) = {} & -\hat{V}_{\sigma,i}(z)\,\hat{V}^{-1}_{\sigma,i}(z)\,\hat{U}_{\sigma,i}(z)\, v^a_{ij}(z) + \hat{U}_{\sigma,i}(z)\, v^a_{ij}(z) \\ & + \hat{V}_{\sigma,i}(z)\, v_{ji}(z) - \hat{V}_{\sigma,i}(z)\, v_{ji}(z) \\ & + \hat{V}_{\sigma,i}(z)\, a_{u,ji}(z) - \hat{U}_{\sigma,i}(z)\, a_{v,ij}(z) - \hat{V}_{\sigma,i}(z)\, a_{v,ji}(z) \\ = {} & \hat{V}_{\sigma,i}(z)\, a_{u,ji}(z) - \hat{U}_{\sigma,i}(z)\, a_{v,ij}(z) - \hat{V}_{\sigma,i}(z)\, a_{v,ji}(z) \end{aligned} \tag{5.59}$$

As a result, if we take $r^\sigma_{u,ij}(z)$ as the diagnostic signal, then we have

Theorem 5.1. *Consider the model of MASs (Eq.5.16) with the consensus-based output feedback control law (Eq.5.18) (equivalently as an FTC controller (Eq.5.24)) shown in Figure 5.5, the attacks $a_{u,ji}$, $a_{v,ij}$ and $a_{v,ji}$ are defined as stealthy, if and only if the condition,*

$$\hat{V}_{\sigma,i}(z)\, a_{u,ji}(z) - \hat{U}_{\sigma,i}(z)\, a_{v,ij}(z) - \hat{V}_{\sigma,i}(z)\, a_{v,ji}(z) = 0, \tag{5.60}$$

is satisfied.

According to Theorem 5.1, there exist two ways to fulfill the stealthy requirement (Eq.5.60).

- if adversaries design $a_{u,ji}(z) = 0$, then the condition,

$$\hat{U}_{\sigma,i}(z)\, a_{v,ij}(z) + \hat{V}_{\sigma,i}(z)\, a_{v,ji}(z) = 0, \tag{5.61}$$
$$a_{v,ji}(z) = -\hat{V}_{\sigma,i}^{-1}(z)\, \hat{U}_{\sigma,i}(z)\, a_{v,ij}(z), \tag{5.62}$$

  should be satisfied.

- on the other hand, when $a_{u,ji}(z)$ is designed as

$$a_{u,ji}(z) = \hat{V}_{\sigma,i}^{-1}(z)\, \hat{U}_{\sigma,i}(z)\, a_{v,ij}(z) + a_{v,ji}(z), \tag{5.63}$$

  the stealthy condition could be also fulfilled.

However, designing the three attacks to keep stealthy is almost impossible. The reason is described as follows.

Assumption 5.5. *We assume that the attackers have no system knowledge, and could not identify a model of an agent without the input and output signals. Furthermore, the attackers have no idea about the purpose of using and transmission $u_{\sigma,ji}$. And the more significant issue is that we keep the switching law confidentiality to attackers.*

Besides $r_{u,ij}^{\sigma}(z)$, the residual signal $r_{0K,i}$ generated by the post-filter of agent i could also be taken into account as the diagnostic signal. As mentioned in Chapter 2, combined with $r_{0K,i}$, not only the cyber-attacks, but also the local faults occurring on agent i could be detected.

To this end, we will discuss about the real-time implementation of the encrypted system, and design of test statistic with a proper threshold in next subsection.

5.2.2 Implementation of the detection and control scheme

In this subsection, the realization of the distributed detection scheme proposed in the last subsection is described step by step.

Without loss of generality, we also take agents i and j as an example. The outputs $y_i(k)$ and $y_j(k)$ combining with measurement noises are first generated on agent i and j, respectively. Through the gain matrix K_α, the reference signals $v_{ij}(k) = K_\alpha y_i(k)$ and $v_{ji}(k) = K_\alpha y_j(k)$ are obtained, and then transmitted between the

two agents with each other via a communication network. In this case, the reference signals are corrupted by cyber-attacks, which have to be diagnosed by the detection scheme.

Next, the encoded signal $u_{\sigma,ji}(k)$ is computed by the encoder on agent j, based on an observer-based residual generator and a feed-forward controller. It is worth mentioning that the parameters of the state feedback gain matrix $F_{\sigma,i}$ and observer gain matrix $L_{\sigma,i}$ switch with time-trigger. For running the encrypted system, the encoded signal is sent from agent j to agent i for detection purposes. As the same with the reference signals, the encoded signal could also be injected with attack signals, $a_{u,ji}(k)$.

On the side of agent i, the diagnostic signal $r^{\sigma}_{u,ij}(k)$ is calculated via a decoder by using three signals as inputs, i.e. the encoded signal $u^{a}_{\sigma,ji}(k)$, the reference signal $v^{a}_{ji}(k)$ and the reference signal $v_{ij}(k)$. It is obviously that the first two inputs of the decoder are received from agent j, and corrupted by cyber-attacks.

Depending on the two diagnostic signals, $r^{\sigma}_{u,ij}(k)$ and the residual signal $r_{0K,i}$ from the local post-filter, we set the test statistic equal to

$$J_{ij}(k) = \lambda \left(r^{\sigma}_{u,ji}(k)\right)^{T} r^{\sigma}_{u,ji}(k) \; + r^{T}_{0K,i}(k)\, \Sigma_r^{-1} r_{0K,i}(k) \tag{5.64}$$

where $\lambda > 0$ is a sufficiently large number.

Recall Eq.5.59, it is remarkable that the diagnostic vector $r^{\sigma}_{u,ji}(k)$ theoretically equals to a zero-mean random vector in the attack-free operation. However, in practice, $r^{\sigma}_{u,ji}(k)$ could be treated as a quasi-random vector with a covariance matrix whose inverse is approximated by λI. Since $r_{0K,i}(k)$ is a white Gaussian noise with $N(0, \Sigma_r)$ distribution, the test statistic

$$J_{ij}(k) \sim \chi^2(n_y) \tag{5.65}$$

is subject to χ^2 distribution with n_y degrees of freedom in an attack-free case. So that, we can design the proper threshold

$$J_{th} = \chi^2_{\alpha}(m) \tag{5.66}$$

with α denoting the upper-bound of FAR.

However, if the agent i or agent j is corrupted by adversaries, then the test

statistic $J_{ij}(k)$ holds

$$J_{ij}(k) = \lambda \left(r^{\sigma}_{u,ji}(k)\right)^{T} r^{\sigma}_{u,ji}(k) \ + r^{T}_{0K,i}(k)\, \Sigma_r^{-1} r_{0K,i}(k) \sim \chi^2(\delta, m) \tag{5.67}$$

where $\chi^2(\delta, n_y)$ denotes a non-central χ^2 distribution with

$$\delta(k) = \lambda \left(r^{\sigma}_{u,ji}(k)\right)^{T} r^{\sigma}_{u,ji}(k) \tag{5.68}$$

To this end, the detection logic can be described as

$$\begin{cases} J_{ij}(k) \leqslant J_{th}, attack - free\ case \\ J_{ij}(k) > J_{th}, attack\ case \end{cases} \tag{5.69}$$

It can be seen that all attacks, $a_{v,ij}$, $a_{v,ji}$ and $a_{u,ji}$, can be well detected as long as adversaries do not identify the system dynamics successfully or know the function of the encrypted detection system.

To be clarity, the distributed detection scheme is summarized step by step in Algorithm 7, which will be verified later in simulation and experimental studies.

Once the i^{th} agent collects all the reference signals from all its neighbors, the distributed detection is implemented corresponding to each of its neighbor. With the detection results,

- in an attack-free case, the control signal $u_i^a(k)$ is generated by the FTC configuration to achieve the consensus goal.
- if the detection results show that at least one of the reference signals from the neighbors of the i^{th} agent is corrupted by a cyber-attack, then the agent should notify the other agents and immediately disconnected from the network.

Algorithm 7 A distributed detection scheme with an encrypted system

Step 1: Outputs measurement and reference signals generation

1: $y_i(k)$ and $y_j(k)$ measurement by local displacement sensors
2: reference signals $v_{ij}(k)$ and $v_{ji}(k)$ generation $v_{ij}(k) = K_\alpha y_i(k),\ v_{ji}(k) = K_\alpha y_j(k)$
3: $v_{ij}(k)$ transmission from agent i to agent j
4: transferring $v_{ji}(k)$ from agent j to agent i

Step 2: Reference signals transmitting interruption and corruption by attackers

1: attack signal $a_{v,ij}(k)$ adding into $v_{ij}(k)$ as $v_{ij}^a(k) = v_{ij}(k) + a_{v,ij}(k)$
2: $v_{ij}(k)$ involving attack signal $a_{v,ij}(k)$ and becoming $v_{ji}^a(k) = v_{ji}(k) + a_{v,ji}(k)$
3: the j^{th} agent receiving $v_{ij}^a(k)$ instead of $v_{ij}(k)$
4: replacing $v_{ji}(k)$, the i^{th} agent obtaining $v_{ji}^a(k)$ as the attacked reference signal

Step 3: Encoder signal $u_{\sigma,ji}(z)$ generation on agent j

1: residual signal $r_{\sigma,ji}(k)$ computation (Eq.5.48) by using the received signal $v_{ij}^a(k)$
2: feed-forward signal $r_{\sigma,ji}(k)$ calculation Eq.5.49 via the reference signal $v_{ji}(k)$
3: encoder signal $u_{\sigma,ji}(k)$ generation (Eq.5.51)
4: the encoder signal transmission from agent j to agent i

Step 4: Encoder signals communication breakdown and hacking by adversaries

1: attack signal $a_{u,ji}(k)$ injection into encoder signal $u_{\sigma,ji}(k)$ so that $u_{\sigma,ji}^a(k) = u_{\sigma,ji}(k) + a_{u,ji}(k)$
2: the i^{th} agent receiving $u_{\sigma,ji}^a(k)$ as the attacked encoder signal

Step 5: Decoder implementation and diagnostic signals generation on agent i

1: one of the two diagnostic signals $r_{u,ij}^\sigma(z)$ delivered by the decoder (Eq.5.56)
2: another diagnostic signal $r_{0K,i}(k)$ generation by the standard observer of agent i

Step 6: Detection logic on agent i

1: test statistic $J_{ij}(k)$ computation combined with $r_{u,ij}^\sigma(z)$, $r_{0K,i}(k)$ and a proper λ
2: cyber-attacks detection by a threshold setting with a ideal FAR and a detection logic

$$\begin{cases} J_{ij}(k) \leqslant J_{th}, attack - free\ case \\ J_{ij}(k) > J_{th}, attack\ case \end{cases}$$

5.3 Simulation Study

5.3.1 Modeling and control configuration of the multi-Robotino system

We consider a five-robot system in our simulation study. The corresponding connected graph of the system is shown in Figure 5.6.

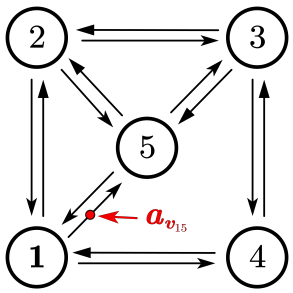

Figure 5.6: Sketch of graph for a multi-agent system with cyber-attack

With the Adjacent matrix $\mathcal{A}$ and Degree matrix $\mathcal{D}$ of the graph, i.e.

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

we can derive the corresponding Laplacian matrix of the graph in our simulation study as follows:

$$\mathcal{L}_{\mathcal{G}} = \mathcal{D} - \mathcal{A} = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & -1 & -1 & 0 & 3 \end{bmatrix} \tag{5.70}$$

In the graph sketched in Figure 5.6, each node denotes a omnidirectional robot called Robotino. We represent a discrete LTI state-space model for every single robot by

$$x_i(k+1) = x_i(k) + T_s u_i(k) + \omega_i(k) \tag{5.71}$$

$$y_i(k) = x_i(k) + \nu_i(k), \quad i = 1, \cdots, 5, \tag{5.72}$$

with $x_i \in \mathcal{R}^1$ denoting the state of robot i, which is the displacement along X_R axis in the robot coordinate. $A = 1$, $B = T_s$, $C = 1$, $D = 0$ with the same value, represent the system matrices for each robot, where $T_s = 0.1(\text{sec})$ indicates the sampling time. $\omega_i(k) \sim N(0, \Sigma_\omega)$ and $\nu_i(k) \sim N(0, \Sigma_\nu)$ are the process noise and measurement noise, respectively. In our simulation study, we set the variance of the noises as, $\Sigma_\omega = 2E-7$, $\Sigma_\nu = 1E-6$.

Therefore, the state-space representation of the five-robot-system is built by stacking all the models of single-robot (Eq.5.71) together. We have

$$x_{\mathcal{G}}(k+1) = A_{\mathcal{G}} x_{\mathcal{G}}(k) + B_{\mathcal{G}} u_{\mathcal{G}}(k) + \omega_{\mathcal{G}}(k) \tag{5.73}$$

$$y_{\mathcal{G}}(k+1) = C_{\mathcal{G}} x_{\mathcal{G}}(k) + \nu_{\mathcal{G}}(k) \tag{5.74}$$

where $\omega_{\mathcal{G}}(k) = \left[\omega_1(k), \cdots, \omega_5(k)\right]^T$ here represents process noise vector, and $\nu_{\mathcal{G}}(k) = \left[\nu_1(k), \cdots, \nu_5(k)\right]^T$ denotes measurement noise vector. The initial condition of the state vector $x_{\mathcal{G}}(k) = \left[x_1(k), \cdots, x_5(k)\right]^T$ is set to $[0, 0.25, 0.5, 0.75, 1]$. The output vector of the multi-Robotino system is defined as $y_{\mathcal{G}}(k) = \left[y_1(k), \cdots, y_5(k)\right]^T$. And the system matrices are described as follows: $A_{\mathcal{G}} = C_{\mathcal{G}} = diag(1,1,1,1,1)$ and $B_{\mathcal{G}} = diag(0.1, 0.1, 0.1, 0.1, 0.1)$.

Then, we could design the consensus-based fault-tolerant controller of the multi-robot system to fulfill the requirement of displacement consensus of all the robots. It is clear that, based on the control law (Eq.5.18), all the agents in the five-robot system are interconnected. With the aid of the Laplacian matrix $\mathcal{L}_{\mathcal{G}}$ and Kronecker product $\otimes$, we can simplify and derive the control vector $u_{\mathcal{G}}(k)$ as

$$u_{\mathcal{G}}(k) = -K_\alpha \left(\mathcal{L}_{\mathcal{G}} \otimes I_{n_x}\right) y_{\mathcal{G}}(k) \tag{5.75}$$

Then, we substitute the consensus control law (Eq.5.75) into Eq.5.16, the following state-space realization of the multi-agent system can be extended as

$$x_{\mathcal{G}}(k+1) = \tilde{A}_{\mathcal{G}} x_{\mathcal{G}}(k) + \omega_{\mathcal{G}}(k) + E_{\mathcal{G}} \nu_{\mathcal{G}}(k) \tag{5.76}$$

where

$$\tilde{A}_{\mathcal{G}} = A_{\mathcal{G}} + \mathcal{L}_{\mathcal{G}} \otimes (BK_{\alpha}C)$$
$$E_{\mathcal{G}} = \mathcal{L}_{\mathcal{G}} \otimes (BK_{\alpha})$$

With setting the proper value of K_{α}, we should guarantee $\tilde{A}_{\mathcal{G}}$ and $A + BF_{0,i},\ i = 1, \cdots, 5$ are Schur matrices by checking the eigenvalues. In our study case, we finally choose $K_{\alpha} = -0.0636$ by using the tuning method, and the local state feedback gain for each agent can be computed and illustrated in Table 5.1.

Table 5.1: Initial condition and local state feedback gain of each agent

Agent i	1	2	3	4	5
$x_i(0)$	0	0.25	0.5	0.75	1
$F_{0,i}$	-0.1909	-0.2545	-0.1909	-0.1273	-0.1909

Next, based on the model description (Eq.5.71) of a single robot, a Luenberger observer-based residual generator of agent i is proposed to obtain the local state x_i and the residual signal $r_{0,i}$ for state feedback control.

$$\begin{cases} \hat{x}_i(k+1) = \hat{x}_i(k) + T_s u_i(k) + L r_{0,i}(k) \\ \hat{y}_i(k) = \hat{x}_i(k) \\ r_{0,i}(k) = y_i(k) - \hat{y}_i(k), \ i = 1, \cdots, 5 \end{cases} \tag{5.77}$$

here, we set the observer gain $L = 1.8$ to make $A - LC$ as a Schur matrix. Then, given the following local observer-based state feedback controller as

$$u_i(z) = F_{0,i}\hat{x}_i(z) - Q(z) r_{0,i}(z) + \bar{v}_i(z),\ i = 1, \cdots, 5 \tag{5.78}$$

Where $Q = 0.01$ is designed to compensate for model uncertainty and disturbance in actual experimental studies.

Figure 5.7 sketches the consensus control performance of the 5-robot system. The results show that the displacement of all the five robots arrives $0.5m$ at about $25sec$, which is the average position for all the agents. Therefore, the effectiveness of the consensus-based FTC configuration is validated by the simulation study.

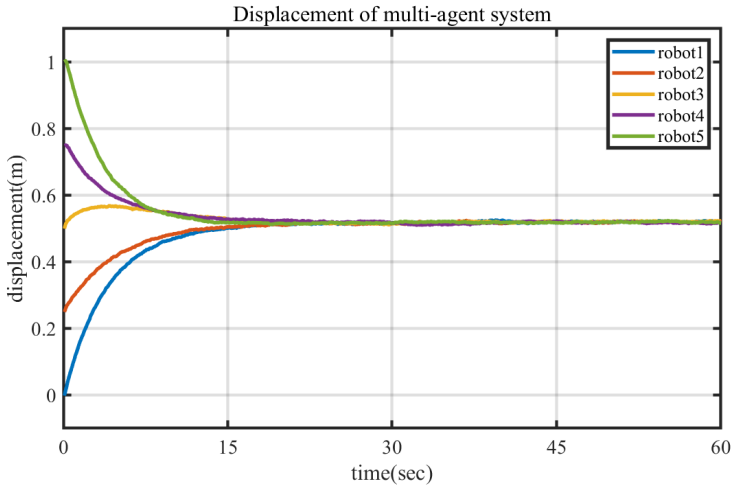

Figure 5.7: Displacement consensus control performance without cyber-attack

5.3.2 Simulation validation of the cyber-attacks detection algorithm

For investigation of the problem on detecting cyber-attacks on the cooperative 5-robot system with a displacement consensus control configuration, as shown in Figure 5.6, an attack signal $a_{v_{15}}(k)$ is injected into the reference signal $v_{15}(k)$, which is transmitted from agent 1 to agent 5. We define the value and the time interval of cyber-attack $a_{v_{15}}(k)$ as follows:

$$\begin{cases} a_{v_{15}} = 0 & t \leqslant t_f \quad attack - free \\ a_{v_{15}} = 0.04 & t > t_f \quad attack\ case \end{cases}$$

where $t_f = 30sec$ is the start time of cyber-attack.

Taking the cyber-attack into account, Figure 5.8 illustrates clearly that, the displacement consensus performance of the multi-Robotino system can hardly be any longer guaranteed. The displacement of the 5^{th} robot is significantly increased, because of the corruption of the reference signal $v_{15}(k)$ by attack $a_{v_{15}}$ at $t \geqslant t_f$. What's more interesting is that not only does robot 5 shift away from its original convergent position under a cyber-attack, but other robots follow the attacked robot slowly changing their steady positions as well.

Simulation results indicate that, the consensus cooperation task of MASs may be compromised, once any agent is attacked by adversaries. It prompts us to set up a detector on each agent distributively to check whether the reference signal (no matter sent to or received from its neighbor) is corrupted by cyber-attacks or not.

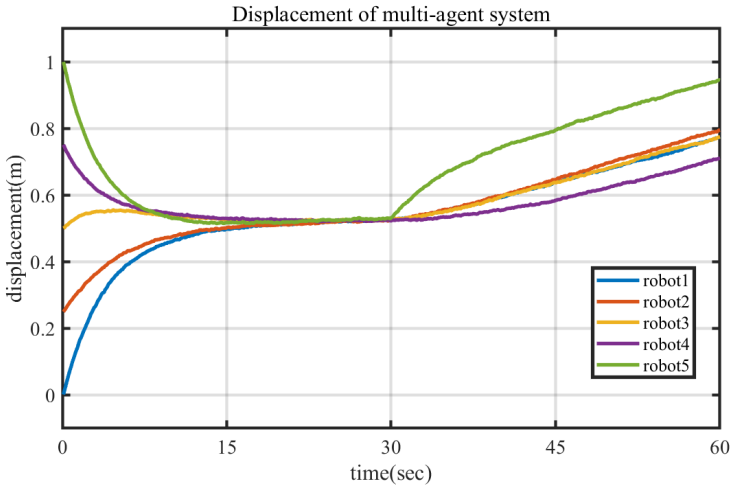

Figure 5.8: Sketch of consensus control performance considering cyber-attack

Test detection by using the standard residual signal

Facing the detection problem of cyber-attack, $a_{v,15}$, the standard observer-based detection scheme for MASs is demonstrated by simulation results. Based on the residual signal generated by a local standard observer and a post-filter, the test detection result is illustrated in Figure 5.9.

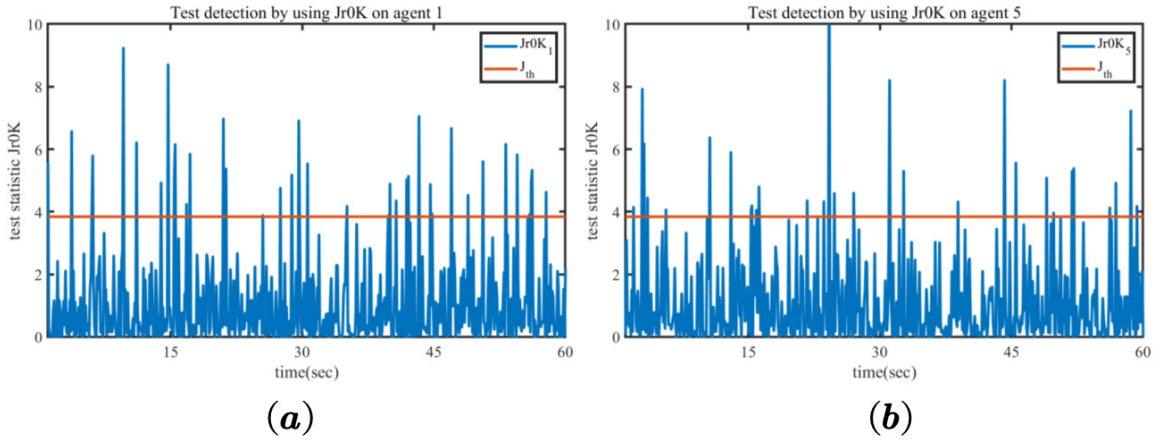

Figure 5.9: Test detection of cyber-attacks delivered by a standard observer

It is clear to see that, although attack signal $a_{v,15}(k)$ is injected into the corresponding reference signal $v_{15}(k)$ via the data transmission from agent 1 to agent 5 during the time interval $t \geqslant t_f$, both of the test statistics, $J_{r0K,1}$ and $J_{r0K,5}$, computed on the standard observer-based detectors on agent 1 and agent 5 are below the threshold J_{th}. It means that the detectors by using a standard residual signal fail to detect the cyber-attacks. The reason is that the residual signal delivered by the standard observer on the attacked agent

remains zero mean, which is unable to reflect the effect of attack signal $a_{v,15}(k)$. To this end, we can regard such type of cyber-attack as a stealthy integrity attack.

Simulation validation of the distributed encrypted switching detection algorithm

Since the standard residual-based detector leads to failure of detection cyber-attacks on the consensus-based FTC configuration of MASs, the distributed encrypted switching detection algorithm is applied to handle the problem. To be specific, in this simulation study, the reference signal $v_{15}(k)$ is corrupted by the attack signal $a_{v,15}(k)$, as shown in Figure 5.6.

To construct the encrypted system including an encoder and a decoder, the switching

Table 5.2: Switched state feedback gains

Agent i	1	2	3	4	5
$F_{1,i}$	-0.2939	-0.2939	-0.2939	-0.1959	-0.2939
$F_{2,i}$	-0.5878	-0.5878	-0.5878	-0.3919	-0.5878
$F_{3,i}$	-0.7837	-0.7837	-0.7837	-0.5225	-0.7837
$F_{4,i}$	-0.5290	-0.5290	-0.5290	-0.3527	-0.5290

Table 5.3: Switched observer gains

σ	1	2	3	4
L_σ	0.9	0.95	1	1.05

state feedback gains $F_{\sigma,i}$ of agent i are designed to satisfy the condition below

$$F_{\sigma,i} \in \mathcal{F} := \{A + BF_{\sigma,i} \; is \; Schur, \sigma \in \mathcal{I}\}, \mathcal{I} = \{1, ..., 4\}$$

The values of the switching state feedback gains for each agent are given in Table 5.2

Besides the switching state feedback gains, the switching observer gains L_σ are also designed to meet the requirements as

$$L_\sigma \in \mathcal{L} := \{A - L_\sigma C \; is \; Schur, \sigma \in \mathcal{I}\}, \mathcal{I} = \{1, ..., 4\}$$

As shown in Table 5.3, the encrypted switching observer gains L_σ for each agent are presented.

The simulation duration is 60 sec. For example, the encoder running at the j^{th} agent generates the encoding signal $u^{\sigma}_{ji}(k)$, while the decoder with the diagnostic signal $r^{\sigma}_{u,ij}(k)$ runs at the i^{th} agent. The switching of encoder and decoder parameters is time-triggered, and the law of switching is described in Table 5.4.

Table 5.4: Switching law for case study

Time interval	0-26sec	26-36sec	36-48sec	48-60sec
Encoder	u^1_{ji}	u^2_{ji}	u^3_{ji}	u^4_{ji}
Decoder	$r^1_{u,ij}$	$r^2_{u,ij}$	$r^3_{u,ij}$	$r^4_{u,ij}$

The distributed detection of cyber-attacks on the 5-robot system is developed in the simulation studies by using computation of test statistic with $\lambda = 20000$ and the threshold setting.

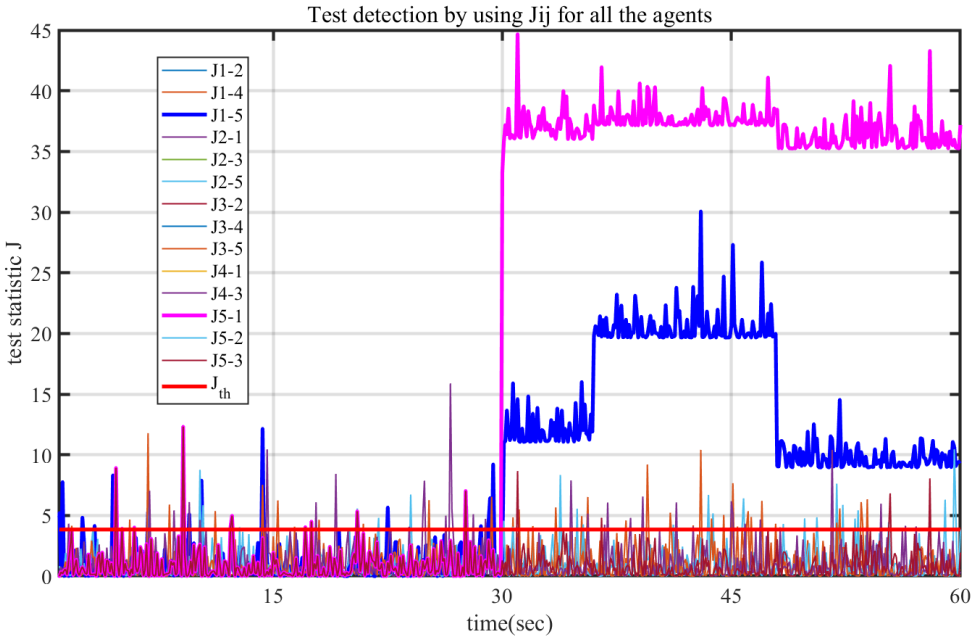

Figure 5.10: Test result of the distributed encrypted detection algorithm

The test statistics of all the detectors are displayed in Figure 5.10. When the attack signal $a_{v,15}$ is injected into the system, only $J_{51}(k)$ (marked pink) and $J_{15}(k)$ (marked dark blue) clearly leap above the threshold. The results revel that the cyber-attack $a_{v,15}$ is detected, by both of the detector operating on agent 5 and the detector implanted on agent 1. As the result, the detection algorithm has been validated by the simulation studies.

5.4 Experimental Study

In this section, an experimental investigation on detecting cyber-attacks on a two-Robotino system is presented.

5.4.1 Experimental system setup

To be more explicit, the experimental setup in this thesis is divided into three parts, as shown in Figure 5.11.

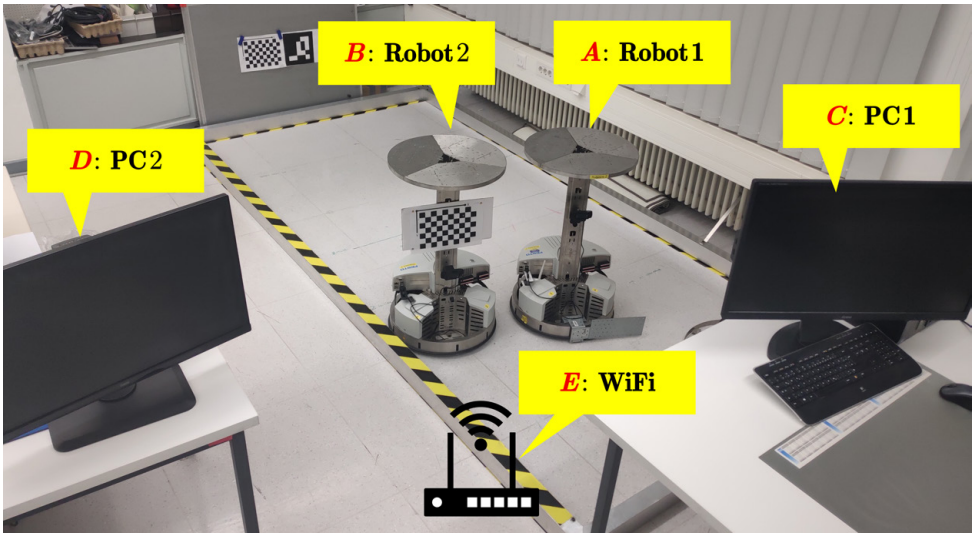

Figure 5.11: Photo of the test setup

- robot 1 and robot 2 are two wheeled robots from the Festo company named the Robotino V3, marked by A and B, respectively.
- two personal computers (PCs) equipped wireless LAN adapters are marked C and D.
- a WiFi marked by E, provides a wireless network.

Thanks to the WiFi provider, the PCs and the robots can be connected wirelessly for remote control using WLAN, and the two PCs may communicate with one another via a wireless link. Figure 5.12 shows the detailed experimental arrangement, with robot 1 and PC 1 forming robot system 1, and robot 2 and PC 2 comprising robot system 2. Following instructions are required for network communications

- in an attack-free scenario, communication is managed by the two PCs themselves.
- If the system is hacked, the attackers may disrupt communication between the two computers and modify data flows.

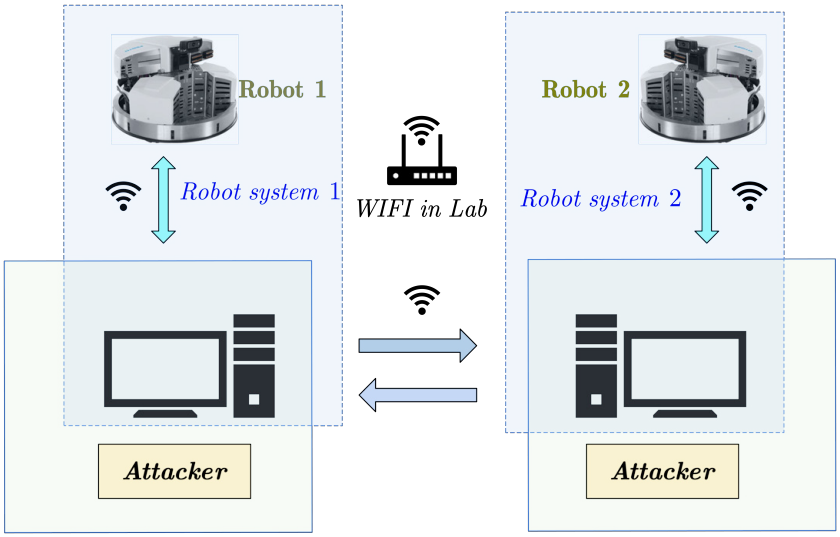

Figure 5.12: Experimental configuration

5.4.2 Communication module design

The communication module in our experiment primarily comprises of two aspects: i) communication between the Robotino and the PC; and ii) data transmission between the two PCs.

Communication between the PC and the robot

We must construct the data transmission module to fulfill our communication requirements in order to transmit the control signal u_i and the measurement data y_i between the PC and the Robotino. The communication function between the PC and the Robotino, on the other hand, has already been coded in the Festo API2 MATLAB tool box. The following are the corresponding MATLAB codes:

- building of the communication connection's initialization

$$>> ComId = Com\_construct;$$

- set the IP address of the matching Robotino.

$$>> Com\_setAddress(ComId, Ip\_address);$$

Here we set "192.168.1.14" for Robotino 1, and set "192.168.1.15" for Robotino 2.

- create a network connection with the relevant Robotino

$$>> Com\_connect(ComId);$$

- termination of the associated Robotino's network connection

$$>> Com\_disconnect(ComId);$$

After that, if connectivity is established successfully, the relevant PC and Robotino can automatically exchange data. As an example, robot system 1 (PC1 and Robotino 1) will be discussed.

- using the following Matlab code, PC 1 can send a control signal to the actuators (DC motors) of Robotino 1.

$$>> OmniDrive\_setVelocity(ComId, v_x, v_y, v_r);$$

- the following Matlab code can be implemented on PC1 to receive measurement data from Robotino 1.

$$>> [S_x, S_y, phi] = Odometry\_get(ComId);$$

Communication between the two PCs

As illustrated in Figure 5.13, we now concentrate on developing the communication module between the two PCs. To be specific,

- the communication is handled by the two PCs themselves in attack-free case;
- the data flow between the two PCs is modified by the attackers if it is under cyber-attack.

For the purpose of simplification, the attackers' modules are programming implemented on the two PCs shown in Figure 5.13. The transmission control protocol (TCP), which is implemented by the communication toolbox in MATLAB, is chosen as the more reliable communication protocol in the experimental investigation. To be specific, PC 1 is configured as a communication server, whereas PC 2 is set as a client. The follows are some advantages of TCP:

- TCP allows for end-to-end communication between the server and the client;

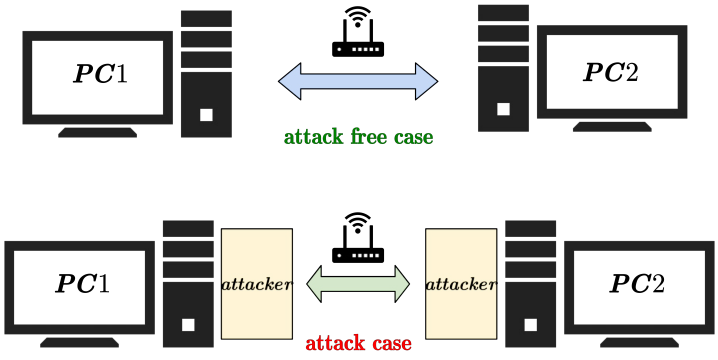

Figure 5.13: Diagram of communication among the two PCs

- the server and client must remain connected during the communication interval to ensure the connection's reliability.

The *tcpserver* and *tcpclient* functions are used to establish a network connection between PC 1 and PC 2, and the emphread() and emphwrite() methods are applied to read and write communication sockets simultaneously.

In Figure 5.14, which is the foundation for TCP communication on MATLAB, we briefly show how TCP socket transmission works.

- firstly, the server creates a 'listening' socket to listen for client connections. The connection can be fully formed using three-way handshakes after the client joins and the server accepts.

  The MATLAB programs for starting the server are as follows:

  $$>> t\_server = tcpip('192.168.1.11', 30000, 'NetworkRole', 'server');$$
  $$>> fopen(t\_server);$$

  The following are the MATLAB codes for opening the client:

  $$>> t\_client = tcpip('192.168.1.10', 30000, 'NetworkRole', 'client');$$
  $$>> fopen(t\_client);$$

- Next, using the emphwrite() and emphread() functions, servers and clients may communicate with one another. The MATLAB programs for the server transmitting

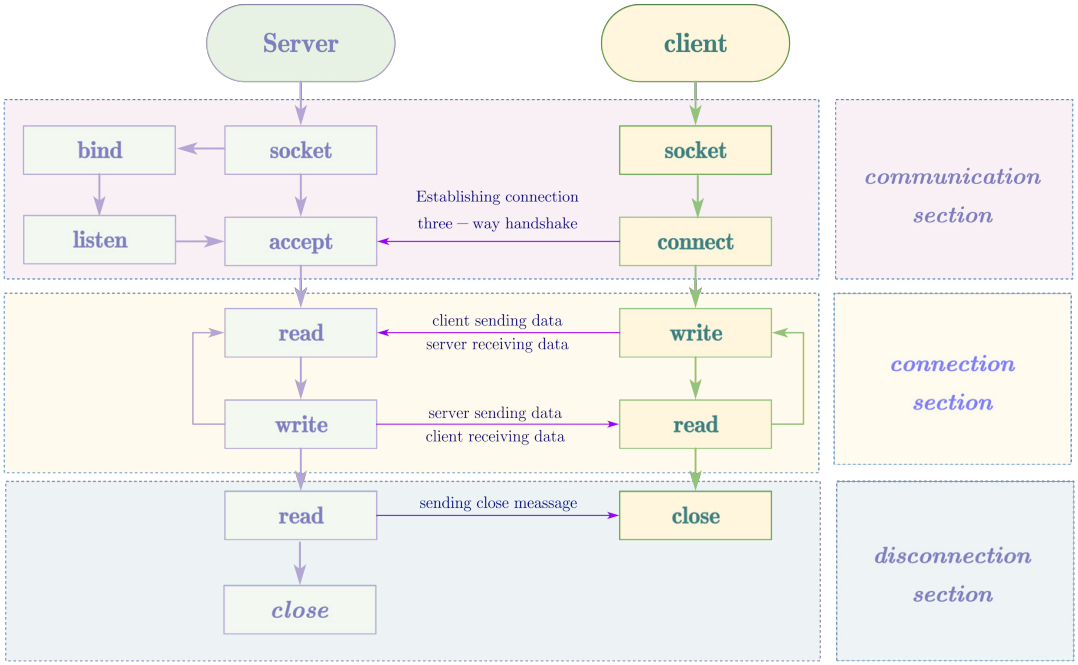

Figure 5.14: Diagram of TCP socket flow

data to the client are as follows:

$$>> fwrite(t\_server, data,' double');$$
$$>> fread(t\_client, datasave,' double');$$

The server transmits the data to the client, who saves it in the variable named 'datasave'.

- finally, by sending close messages from the client to the server, the communication can be terminated. The following are the MATLAB programs for disconnecting the server and client:

$$>> fclose(t\_server);$$
$$>> fclose(t\_client);$$

The cyber-attack detection of consensus-based FTC for MASs can be validated experimentally later using TCP communication protocol and MATLAB functions.

5.4.3 Consensus control with real-time communication

In the experimental studies, the control purpose is to obtain displacement consensus between the two robots, as shown in Figure 5.15. The Adjacent matrix and Degree matrix of the graph are constructed in the same way as in the simulation study, using the graph theory presented in Chapter 2.

$$\mathcal{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The corresponding Laplace matrix can therefore be calculated as

$$\mathcal{L}_{\mathcal{G}} = \mathcal{D} - \mathcal{A} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Remember that each omnidirectional wheeled robot's LTI model is identical to the model (Eq.5.71) used in the simulation research. The system matrices for each robot are $A = 1$, $B = Ts$, $C = 1$, $D = 0$, with $T_s = 0.1(\text{sec})$ signifying the sampling time. Combine the models of both two agents (Eq.5.71), It holds

$$x_{\mathcal{G}}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_{\mathcal{G}}(k) + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} u_{\mathcal{G}}(k) + \omega_{\mathcal{G}}(k)$$

with $\omega_{\mathcal{G}}(k)$ here representing process noise vector. The initial states of the two robots are $x_1(0) = 0m$, $x_2(0) = 1m$. Then, according to the consensus control law, we design the state feedback gain as $F_{0,1} = F_{0,2} = -0.1295$.

The process noise vector is represented by $\omega_{\mathcal{G}}(k)$. The two robots' initial states are $x_1(0) = 0m$, $x_2(0) = 1m$. The state feedback gain is therefore designed as $F_{0,1} = F_{0,2} = -0.1295$ according to the consensus control law. Following that, we develop a Luenberger observer-based residual generator of agent i based on the model description (Eq.5.71) of a single robot in order to acquire the local state x_i and the residual signal $r_{0,i}$, $i = 1, 2$.

$$\begin{cases} \hat{x}_i(k+1) = \hat{x}_i(k) + T_s u_i(k) + L r_{0,i}(k) \\ \hat{y}_i(k) = \hat{x}_i(k) \\ r_{0,i}(k) = y_i(k) - \hat{y}_i(k), \ i = 1, 2 \end{cases}$$

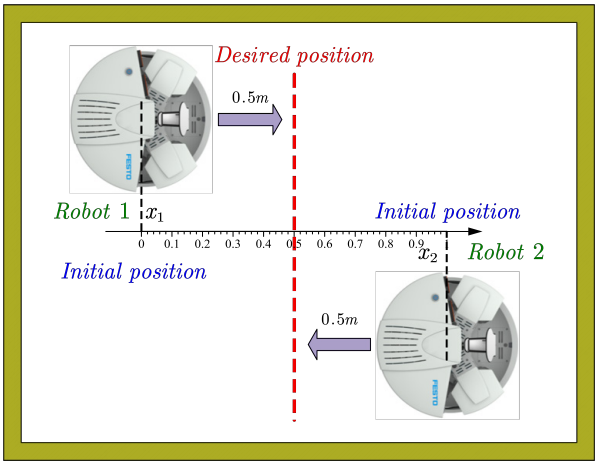

Figure 5.15: Sketch of the experimental task description

To make $A - LC$ a Schur matrix, we set the observer gain to $L = 1.8$. The initial estimate states are the same as the initial states of the two robots, i.e.

$$\hat{x}_1(0) = x_1(0),\ \hat{x}_2(0) = x_2(0)$$

The observer-based state feedback control law in our experiment can thus be written as follows:

$$u_1(k) = F_{0,1}\hat{x}_1(k) - Qr_{0,1}(k) + \bar{v}_1(k)$$
$$u_2(k) = F_{0,2}\hat{x}_2(k) - Qr_{0,1}(k) + \bar{v}_2(k)$$

To enhance the robustness of a single robot i, the compensation parameter matrix $Q = 0.01$ is computed offline.

The displacement consensus performance of the 2-robot system is depicted in Figure 5.16. The displacement for both agents converges to $0.5m$ for the time span $[0, 30]$sec, demonstrating that the consensus-based FTC configuration established in our experimental investigation fits the requirements of consensus control. The 2-Robotino system reaches a steady state at $t = 20$sec, according to the results.

5.4.4 Validation of attack detection algorithm

When $t = 30$sec, the attacker injects attack signals into both of the two robots in our experiment. To be more explicit, $a_{v,12} = 0.04$ corrupts the reference signal v_{12}, while the attack signal $a_{v_{21}} = -0.04$ injects into the reference signal v_{21} at the same time.

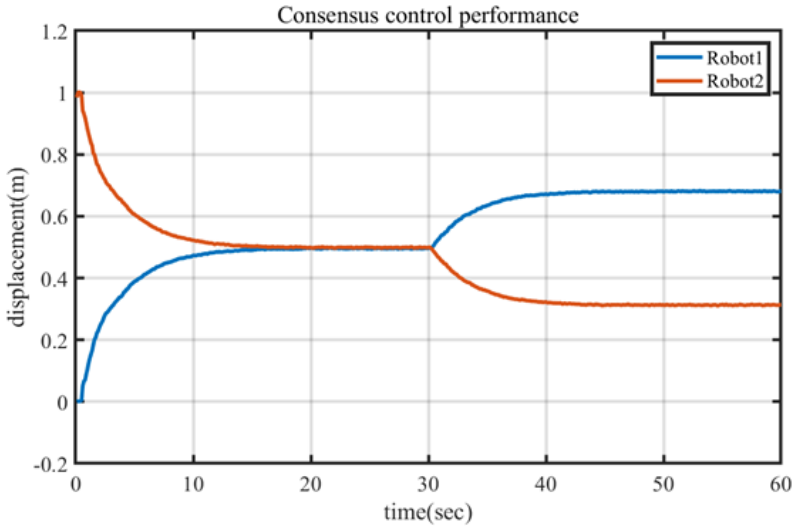

Figure 5.16: Consensus control performance of the two robots

The time interval $[30, 60]sec$ in Figure 5.16 represents the displacement of the two attacked robots. It is clear that the displacement consensus goal is unable to guarantee when the 2-robot-system is corrupted by cyber-attacks. In our case, two wheeled robots enter another steady states of their own, which are different from each other. To this end, the failure of the consensus control task motivates us to propose a detection scheme to diagnose cyber-attacks.

In order to setup the encrypted encoder and decoder, the switching state feedback gains $F_{\sigma,i}$ and observer-gains $L_{\sigma,i}$ are designed as shown in Table 5.5. It is important to note that while designing $F_{\sigma,i}$ and $L_{\sigma,i}$, where $i = 1, 2$, the matrices $A + BF_{\sigma,i}$ and $A - L_{\sigma,i}C$ should be Schur matrices.

Table 5.5: Switched state feedback gain and observer gain for experimental study

σ	1	2	3	4
$L_{\sigma,i}$	0.9	0.95	1	1.05
$F_{\sigma,i}$	-0.1943	-0.2590	-0.1554	-0.3237

As stated in Table 5.4, we use the same switching law as the simulation research. We further suppose that an attacker cannot obtain the switching state feedback gains, observer gains, or switching law, since the system is encrypted with a switching system.

According to Eq.5.64, the test statistic J_{ij} is able to be generated combining the

two diagnostic signal, i.e. $r^{\sigma}_{u,ij}$ and $r_{0K,i}$. We set the factor $\lambda = 20000$ for computing $r^{\sigma}_{u,ij}$. Based on a proper threshold $J_{th} = 3.8415$ calculated though $FAR = 0.05$ and 1-degree-χ^2 distribution, the distributed detection results is illustrated in Figure 5.17.

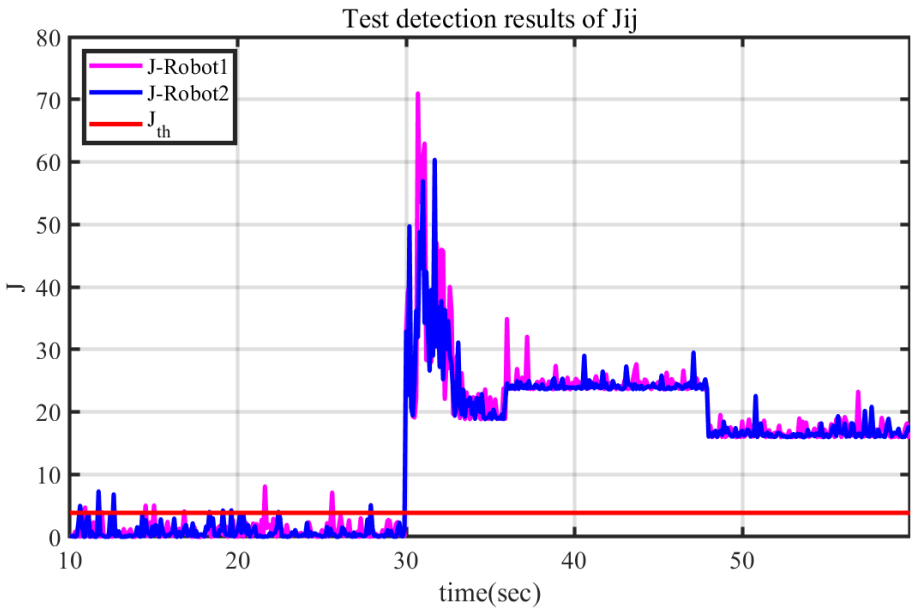

Figure 5.17: Experimental detection results by using encrypted detector

The detection results demonstrated that following the attack signals $a_{v,12}(k)$ and $a_{v,21}(k)$ being injected into the 2-Robotino-system, the test statistics $J_{12}(k)$ and $J_{21}(k)$ both obviously surpass the threshold. It is worth mentioning that, the test statistics in the time interval $[30, 35]sec$ are significantly larger. Because of model uncertainties, the residual signals $r_{0,1}$ and $r_{0,2}$ change rapidly, especially during an unstable state of the robot system. Thus, the cyber-attack signals, $a_{v,12}$ and $a_{v,21}$ can be clearly detected by using the detectors on robot 1 and robot 2, respectively. The experimental results can validate the effectiveness and feasibility of the distributed encrypted detection algorithm proposed in Section 3.4.

5.5 Concluding Remarks

In this chapter, we addressed the distributed detection problem for cooperative MASs under stealthy integrity cyber-attacks.

In the beginning, we attempt to construct a general model of single-agent and multi-agent systems. Based on the graph study, coprime factorization techniques, and parameterization of the stabilizing controller, the configuration of the consensus-based FTC driven by an observer-based residual generator is designed for each agent to achieve

the displacement consensus of MASs. We explain that MASs with this control architecture are vulnerable to cyber-attacks that are hard to detect by using a standard observer-based detector. Therefore, such cyber-attacks can be regarded as stealthy integrated cyber-attack.

Combined with a switching system that includes an encoder and a decoder, a distributed encrypted detection scheme is then proposed to diagnose if the reference signal conveyed between neighbors is under attack or not. It is worth noting that the detector in each agent can hardly affect the original control and monitoring capabilities. On the other hand, the encrypted transmission data prohibits adversaries from capturing information across the communication network to figure out how MASs function.

Finally, to validate the effectiveness and applicability of the proposed distributed stealthy cyber-attack detection algorithm on MASs, not only a series of simulation studies but also experimental studies of a real-time robot system with TCP-based communication are provided.

6 Conclusion and Future works

The main focus of this thesis is on the investigation of distributed observer-based detection schemes for faults and cyber-attacks on multi-agent systems, which are developed for modern industries. Several significant contributions have been made to achieve the final goal. In this chapter, we first conclude the main insights and contributions of the work. Then we discuss and point out new directions, in which further research can be conducted.

6.1 Conclusion

This thesis has been dedicated to the development of distributed state estimation and fault detection on MASs. It integrated theoretical fundamentals and, at the same time, delivered new and practical results that could benefit real-time industrial automatic applications. The thesis has been concluded as follows:

In Chapter 1, we have given a brief introduction to this thesis. It started with the background and motivation of the chosen topic, i.e. "*Distributed Detection of Faults and Cyber-attacks of Multi-Agent Systems* and the research scope of the work in the context of implementation of detection in distributed systems. It included the objective of the work and an outline of the rest of the thesis.

In Chapter 2, we have presented definitions, fundamental knowledge, and mathematical preliminaries of four main aspects, i.e. graph theory, description of dynamic systems, fault detection, and FTC configuration. We have proposed the definition of a multi-layer neighborhood, which is essential for later description of MASs. Using the controller parameterization in the FTC configuration, we can have another interpretation that any output feedback controller is an observer-based controller driven through the residual vector, which is also a core of distributed detection of deception cyber-attacks for MASs.

In Chapter 3, we have dealt with state estimation and small fault detection problems by using large-scale time-varying sensor networks, which have high variance of local measurement noises. We started by establishing models of a target plant and communication

iteration of sensor networks. Then, we have proposed a distributed H_2 observer in a recursive form and a detection scheme, embedded on each sensor node. The performance of estimation and fault detection depending on the proposed observer in Chapter 3 are verified by simulation case studies.

In Chapter 4, we have devoted ourselves to solving the distributed detection problem of DoS cyber-attacks on wireless sensor networks. Firstly, depending on the residual signal generated by the local Kalman filter, we have proposed a communication iteration method to compute the diagnostic signal for each detector operating independently at each sensor node. Secondly, we have pointed out that, the variance of diagnostic signals has unknown changes once some communication links are disrupted by DoS attacks. Thirdly, we have proposed a GLR approach to generate a test statistic for detecting DoS attacks, and an offline statistical training method to achieve the proper threshold. Finally, through simulation studies, the effectiveness and feasibility of the proposed approaches in Chapter 4 are validated.

In Chapter 5, we have addressed the distributed detection problem for cooperative MASs under stealthy integrity cyber-attacks. At first, depending on the graph study, coprime factorization techniques, and parameterization of the stabilizing controller, we have designed the configuration of consensus-based FTC driven by an observer-based residual generator. Then, combining a switching system with both an encoder and a decoder, we have given a distributed encrypted detection scheme to determine whether the reference signals transmitted between neighboring agents are secure from potential attacks. It's important to highlight that the detectors in each agent are carefully designed so as not to interfere with the core control and monitoring performance. Meanwhile, the use of encrypted transmission data serves as a protective barrier, preventing adversaries from eavesdropping on the communication network and gaining insights into how MASs operate. At last, to confirm the efficiency and suitability of the proposed distributed stealthy cyber-attack detection algorithm in Chapter 5, we have presented a comprehensive evaluation that encompasses a range of simulation investigations as well as practical experiments involving a real-time robot system utilizing TCP-based communication.

6.2 Future Works

While this thesis made considerable contributions to distributed detection of faults and cyber-attacks of MASs, the following topics have not been included but worth been explored in the future.

Data driven approaches

In this thesis, all the proposed methods are based on system models. However, large-scale complex MASs can exhibit highly complex and nonlinear behavior, making it challenging to develop traditional model-based approaches, which are designed offline and non-scalable. In this case, a data-driven realization of the proposed methods in Chapters 3, 4, and 5 have several advantages as follows:

- Data-driven models are built based on real-world data, which allows them to accurately represent the underlying behavior and dynamics of MASs. They can capture complex and nonlinear relationships that may be challenging to model analytically.
- Modelling by data-driven approaches can adapt and update in real time as new data becomes available. This adaptability is particularly useful for systems with changing dynamics or external influences. With continuous data collection and model updates, data-driven models can improve over time, ensuring they remain relevant and accurate.
- Data-driven models can handle large volumes of data, making them well-suited for large-scale MASs with numerous agents. They can analyze and process extensive datasets efficiently.
- Data-driven models have predictive power. These approaches can make accurate predictions about future system behavior, which is valuable for decision-making and control in fields like finance, healthcare, and manufacturing.

In summary, data-driven approaches are a valuable choice for modeling large-scale multi-agent systems because they are capable of handling complexity, adapting to changes, and providing real-time insights. These models can help organizations and researchers gain a deeper understanding of and better control over complex multi-agent systems.

Fault tolerant control

This thesis only focuses on detection problems of faults and cyber-attacks of MASs, without considering FTC for further study. However, FTC is crucial for multi-agent systems in the context of both faults and cyber-attacks as follows:

- MASs often perform critical tasks where uninterrupted operation is essential, such as autonomous vehicles, industrial automation, or healthcare robots. When individual agents or components experience hardware failures or environmental issues, FTC ensures that the entire system can continue functioning. This resilience minimizes downtime and service disruptions, maintaining mission-critical functions.

- FTC can optimize how tasks and resources are allocated among agents within the MAS, especially when dealing with faults or cyber-attacks. By reallocating responsibilities and resources dynamically, the system maintains performance levels, even under adverse conditions that one or more agents are disconnected from the original graph. This resource optimization prevents overburdening healthy agents and minimizes performance degradation, contributing to efficient and effective operation.

Bibliography

[1] A. Abdo, R. Ibrahim, and N. A. Rawashdeh, "Mobile robot localization evaluations with visual odometry in varying environments using festo-robotino," 2020.

[2] A. A. Alfantookh, "Dos attacks intelligent detection using neural networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 18, pp. 31–51, 2006.

[3] A. Amirkhani and A. H. Barshooi, "Consensus in multi-agent systems: a review," *Artificial Intelligence Review*, vol. 55, no. 5, pp. 3897–3935, 2022.

[4] X. Bai, Z. Wang, L. Zou, and F. E. Alsaadi, "Collaborative fusion estimation over wireless sensor networks for monitoring co2 concentration in a greenhouse," *Information Fusion*, vol. 42, pp. 119–126, 2018.

[5] G. Bao, L. Ma, and X. Yi, "Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: A survey," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 539–551, 2022.

[6] R. V. Beard, "Failure accomodation in linear systems through self-reorganization." Ph.D. dissertation, Massachusetts Institute of Technology, 1971.

[7] A. Cheema, M. Tariq, A. Hafiz, M. M. Khan, F. Ahmad, and M. Anwar, "Prevention techniques against distributed denial of service attacks in heterogeneous networks: A systematic review," *Security and Communication Networks*, vol. 2022, pp. 1–15, 2022.

[8] X. Chen, K. Zhang, and B. Jiang, "Finite-time unknown input observer-based distributed fault diagnosis for multi-agent systems with disturbances," *Circuits, Systems, and Signal Processing*, vol. 37, pp. 4215–4233, 2018.

[9] Y. Chen, K. Hwang, and W.-S. Ku, "Distributed change-point detection of ddos attacks over multiple network domains," in *Int. Symp. on Collaborative Technologies and Systems*. Citeseer, 2006, pp. 543–550.

[10] Y. Chen, S. Kar, and J. M. Moura, "Resilient distributed estimation through adversary detection," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2455–2469, 2018.

[11] M. R. Davoodi, K. Khorasani, H. A. Talebi, and H. R. Momeni, "Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1061–1069, 2013.

[12] M. Davoodi, N. Meskin, and K. Khorasani, "Simultaneous fault detection and consensus control design for a network of multi-agent systems," *Automatica*, vol. 66, pp. 185–194, 2016.

[13] M. Deghat, V. Ugrinovskii, I. Shames, and C. Langbort, "Detection and mitigation of biasing attacks on distributed estimation networks," *Automatica*, vol. 99, pp. 369–381, 2019.

[14] C. Deng, D. Zhang, and G. Feng, "Resilient practical cooperative output regulation for mass with unknown switching exosystem dynamics under dos attacks," *Automatica*, vol. 139, p. 110172, 2022.

[15] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of cps security," *Annual reviews in control*, vol. 47, pp. 394–411, 2019.

[16] S. X. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools, 2nd Edition*. Springer Science & Business Media, 2008.

[17] ——, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. London: Springer-Verlag, 2013.

[18] ——, *Data-driven design of fault diagnosis and fault-tolerant control systems*. Springer-Verlag, 2014.

[19] ——, *Advanced methods for fault diagnosis and fault-tolerant control*. Springer, 2021.

[20] ——, *Advanced methods for fault diagnosis and fault-tolerant control*. Springer, 2021.

[21] S. X. Ding, L. Li, D. Zhao, C. Louen, and T. Liu, "Application of the unified control and detection framework to detecting stealthy integrity cyber-attacks on feedback control systems," *arXiv preprint arXiv:2103.00210*, 2021.

[22] S. X. Ding, G. Yang, P. Zhang, E. L. Ding, T. Jeinsch, N. Weinhold, and M. Schultalbers, "Feedback control structures, embedded residual signals, and feedback control schemes with an integrated residual access," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 352–367, 2009.

[23] X. Ding and P. M. Frank, "Fault detection via factorization approach," *Systems & control letters*, vol. 14, no. 5, pp. 431–436, 1990.

[24] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *Ieee Access*, vol. 6, pp. 28 573–28 593, 2018.

[25] E. Espina, J. Llanos, C. Burgos-Mellado, R. Cardenas-Dobson, M. Martinez-Gomez, and D. Saez, "Distributed control strategies for microgrids: An overview," *IEEE Access*, vol. 8, pp. 193 412–193 448, 2020.

[26] R. M. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis with overlapping decompositions: An adaptive approximation approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 794–799, 2009.

[27] ——, "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.

[28] N. Forti, G. Battistelli, L. Chisci, S. Li, B. Wang, and B. Sinopoli, "Distributed joint attack detection and secure state estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 96–110, 2017.

[29] A. J. Gallo, M. S. Turan, P. Nahata, F. Boem, T. Parisini, and G. Ferrari-Trecate, "Distributed cyber-attack detection in the secondary control of dc microgrids," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 344–349.

[30] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE transactions on industrial electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[31] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2001, vol. 207.

[32] P. Griffioen, S. Weerakkody, B. Sinopoli, O. Ozel, and Y. Mo, "A tutorial on detecting security attacks on cyber-physical systems," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 979–984.

[33] W. He, Z. Mo, Q.-L. Han, and F. Qian, "Secure impulsive synchronization in lipschitz-type multi-agent systems subject to deception attacks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1326–1334, 2020.

[34] X. He, Z. Wang, Y. Liu, and D.-H. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1670–1679, 2013.

[35] X. He, Z. Wang, and D. Zhou, "Robust fault detection for networked systems with communication delay and data missing," *Automatica*, vol. 45, no. 11, pp. 2634–2639, 2009.

[36] J. P. Hespanha and A. S. Morse, "Stability of switched systems with average dwell-time," in *Proceedings of the 38th IEEE conference on decision and control (Cat. No. 99CH36304)*, vol. 3. IEEE, 1999, pp. 2655–2660.

[37] L. Huang, M. Zhou, K. Hao, and E. Hou, "A survey of multi-robot regular and adversarial patrolling," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 894–903, 2019.

[38] 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.

[39] Y. Jiang, B. Niu, X. Wang, X. Zhao, H. Wang, and B. Yan, "Distributed finite-time consensus tracking control for nonlinear multi-agent systems with fdi attacks and application to single-link robots," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 4, pp. 1505–1509, 2022.

[40] B. Kailkhura, Y. S. Han, S. Brahma, and P. K. Varshney, "Distributed bayesian detection in the presence of byzantine data," *IEEE transactions on signal processing*, vol. 63, no. 19, pp. 5250–5263, 2015.

[41] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," 1961.

[42] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[43] C. Keliris, M. M. Polycarpou, and T. Parisini, "A distributed fault detection filtering approach for a class of interconnected continuous-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 2032–2047, 2013.

[44] S. Kelkar and R. Kamal, "Adaptive fault diagnosis algorithm for controller area network," *IEEE transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5527–5537, 2014.

[45] I. Khalil, J. Doyle, and K. Glover, *Robust and optimal control*. prentice hall, new jersey, 1996.

[46] Y. Lei, Y. Yuan, and J. Zhao, "Model-based detection and monitoring of the intermittent connections for can networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2912–2921, 2013.

[47] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

[48] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017.

[49] Y. Li, P. Zhang, L. Zhang, and B. Wang, "Active synchronous detection of deception attacks in microgrid control systems," *IEEE transactions on smart grid*, vol. 8, no. 1, pp. 373–375, 2016.

[50] B. Liu, W. Han, E. Wang, S. Xiong, L. Wu, Q. Wang, J. Wang, and C. Qiao, "Multi-agent attention double actor-critic framework for intelligent traffic light control in urban scenarios with hybrid traffic," *IEEE Transactions on Mobile Computing*, 2023.

[51] M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, and J. C.-W. Lin, "Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review," *Sensors*, vol. 22, no. 6, p. 2087, 2022.

[52] R. Mathe and K. Folly, "Impact of large scale grid-connected wind generators on the power system network," in *2017 IEEE PES PowerAfrica*. IEEE, 2017, pp. 328–333.

[53] P. P. Menon and C. Edwards, "Robust fault estimation using relative information in linear multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 477–482, 2013.

[54] M. Mercangöz and F. J. Doyle III, "Distributed model predictive control of an experimental four-tank system," *Journal of process control*, vol. 17, no. 3, pp. 297–308, 2007.

[55] M. J. Morshed, "A nonlinear coordinated approach to enhance the transient stability of wind energy-based power systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1087–1097, 2020.

[56] M. J. Morshed and A. Fekih, "A novel fault ride through scheme for hybrid wind/pv power generation systems," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 4, pp. 2427–2436, 2019.

[57] A. Mousavi, K. Aryankia, and R. R. Selmic, "A distributed fdi cyber-attack detection in discrete-time nonlinear multi-agent systems using neural networks," *European Journal of Control*, vol. 66, p. 100646, 2022.

[58] S. Mukkamala and A. H. Sung, "Detecting denial of service attacks using support vector machines," in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03.*, vol. 2. IEEE, 2003, pp. 1231–1236.

[59] X. Nan and X. Xiaowen, "Robot experiment simulation and design based on festo robotino," *2011 IEEE 3rd International Conference on Communication Software and Networks*, pp. 160–162, 2011.

[60] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[61] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 6698–6703.

[62] Y. Pan, Y. Wu, and H.-K. Lam, "Security-based fuzzy control for nonlinear networked control systems with dos attacks via a resilient event-triggered scheme," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 10, pp. 4359–4368, 2022.

[63] S. Patil and S. Chaudhari, "Dos attack prevention technique in wireless sensor networks," *Procedia Computer Science*, vol. 79, pp. 715–721, 2016.

[64] K. Peng, K. Zhang, G. Li, and D. Zhou, "Contribution rate plot for nonlinear quality-related fault diagnosis with application to the hot strip mill process," *Control Engineering Practice*, vol. 21, no. 4, pp. 360–369, 2013.

[65] Z. Peng, D. Wang, Y. Shi, H. Wang, and W. Wang, "Containment control of networked autonomous underwater vehicles with model uncertainty and ocean disturbances guided by multiple leaders," *Information Sciences*, vol. 316, pp. 163–179, 2015.

[66] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2016.

[67] Y. Quan, W. Chen, Z. Wu, and L. Peng, "Distributed fault detection and isolation for leader–follower multi-agent systems with disturbances using observer techniques," *Nonlinear Dynamics*, vol. 93, no. 2, pp. 863–871, 2018.

[68] M. Rajaee and K. Mazlumi, "Multi-agent distributed deep learning algorithm to detect cyber-attacks in distance relays," *IEEE Access*, vol. 11, pp. 10 842–10 849, 2023.

[69] A. Saifullah, "Defending against distributed denial-of-service attacks with weight-fair router throttling," 2009.

[70] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.

[71] J. Shi, X. He, Z. Wang, and D. Zhou, "Distributed fault detection for a class of second-order multi-agent systems: an optimal robust observer approach," *IET Control Theory & Applications*, vol. 8, no. 12, pp. 1032–1044, 2014.

[72] S. Tan, J. M. Guerrero, P. Xie, R. Han, and J. C. Vasquez, "Brief survey on attack detection methods for cyber-physical systems," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5329–5339, 2020.

[73] G. Vinnicombe, *Uncertainty and Feedback: H [infinity] Loop-shaping and the [nu]-gap Metric*. World Scientific, 2001.

[74] H. Wang, D. Zhang, and K. G. Shin, "Change-point monitoring for the detection of dos attacks," *IEEE Transactions on dependable and secure computing*, vol. 1, no. 4, pp. 193–208, 2004.

[75] J. Wang, Y. Hong, J. Wang, J. Xu, Y. Tang, Q.-L. Han, and J. Kurths, "Cooperative and competitive multi-agent systems: From optimization to games," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 763–783, 2022.

[76] Y. Wang, H. Ye, and G. Wang, "Fault detection of ncs based on eigendecomposition, adaptive evaluation and adaptive threshold," *International Journal of Control*, vol. 80, no. 12, pp. 1903–1911, 2007.

[77] G. Wen, X. Zhai, Z. Peng, and A. Rahmani, "Fault-tolerant secure consensus tracking of delayed nonlinear multi-agent systems with deception attacks and uncertain parameters via impulsive control," *Communications in nonlinear science and numerical simulation*, vol. 82, p. 105043, 2020.

[78] C. Wuthishuwong and A. Traechtler, "Distributed control system architecture for balancing and stabilizing traffic in the network of multiple autonomous intersections using feedback consensus and route assignment method," *Complex & Intelligent Systems*, vol. 6, no. 1, pp. 165–187, 2020.

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

[80] W. Yu, G. Wen, G. Chen, and J. Cao, *Distributed cooperative control of multi-agent systems*. John Wiley & Sons, 2017.

[81] D. Zhang, G. Feng, Y. Shi, and D. Srinivasan, "Physical safety and cyber security analysis of multi-agent systems: A survey of recent advances," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 319–333, 2021.

[82] J. Zhang, J. Song, J. Li, F. Han, and H. Zhang, "Observer-based non-fragile hinf-consensus control for multi-agent systems under deception attacks," *International Journal of Systems Science*, vol. 52, no. 6, pp. 1223–1236, 2021.

[83] S.-Q. Zhang, W.-W. Che, and C. Deng, "Observer-based event-triggered secure synchronization control for multi-agent systems under false data injection attacks," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 8, pp. 4843–4860, 2022.

[84] W.-A. Zhang, S. Liu, and L. Yu, "Fusion estimation for sensor networks with nonuniform estimation rates," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 5, pp. 1485–1498, 2014.

[85] W.-A. Zhang, L. Yu, and D. He, "Sequential fusion estimation for sensor networks with deceptive attacks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1829–1843, 2019.

[86] X. Zhang, "Decentralized fault detection for a class of large-scale nonlinear uncertain systems," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 5650–5655.

[87] Y. Zhang, X. Li, and Y. Liu, "The detection and defence of dos attack for wireless sensor network," *The journal of china universities of posts and telecommunications*, vol. 19, pp. 52–56, 2012.

[88] X. Zhao, L. Zhang, P. Shi, and M. Liu, "Stability and stabilization of switched linear systems with mode-dependent average dwell time," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1809–1815, 2011.

[89] Y. Zheng, S. E. Li, K. Li, and W. Ren, "Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed h-infinity controller synthesis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1353–1364, 2017.

[90] D.-H. Zhou and P. Frank, "Fault diagnostics and fault tolerant control," *IEEE Transactions on aerospace and electronic systems*, vol. 34, no. 2, pp. 420–427, 1998.

[91] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[92] K. Zhou and Z. Ren, "A new controller architecture for high performance, robust, and fault-tolerant control," *IEEE Transactions on automatic control*, vol. 46, no. 10, pp. 1613–1618, 2001.

[93] R. Zhu, L. Li, S. Wu, P. Lv, Y. Li, and M. Xu, "Multi-agent broad reinforcement learning for intelligent traffic light control," *Information Sciences*, vol. 619, pp. 509–525, 2023.

[94] Q. Zong, F. Zeng, W. Liu, Y. Ji, and Y. Tao, "Sliding mode observer-based fault detection of distributed networked control systems with time delay," *Circuits, Systems, and Signal Processing*, vol. 31, no. 1, pp. 203–222, 2012.